Large Dimensional Analysis and Improvement of Multi Task Learning

Malik Tiomoko malik.tiomoko@u-psud.fr
Laboratoire des Signaux et Systèmes
University of Paris Saclay
Orsay, France

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

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

Editor:

Abstract

Multi Task Learning (MTL) efficiently leverages useful information contained in multiple related tasks to help improve the generalization performance of all tasks. This article conducts a large dimensional analysis of a simple but, as we shall see, extremely powerful when carefully tuned, Least Square Support Vector Machine (LSSVM) version of MTL, in the regime where the dimension p of the data and their number n grow large at the same rate.

Under mild assumptions on the input data, the theoretical analysis of the MTL-LSSVM algorithm first reveals the "sufficient statistics" exploited by the algorithm and their interaction at work. These results demonstrate, as a striking consequence, that the standard approach to MTL-LSSVM is largely suboptimal, can lead to severe effects of *negative transfer* but that these impairments are easily corrected. These corrections are turned into an improved MTL-LSSVM algorithm which can only benefit from additional data, and the theoretical performance of which is also analyzed.

As evidenced and theoretically sustained in numerous recent works, these large dimensional results are robust to broad ranges of data distributions, which our present experiments corroborate. Specifically, the article reports a systematically close behavior between theoretical and empirical performances on popular datasets, which is strongly suggestive of the applicability of the proposed carefully tuned MTL-LSSVM method to real data. This fine-tuning is fully based on the theoretical analysis and does not in particular require any cross validation procedure. Besides, the reported performances on real datasets almost systematically outperform much more elaborate and less intuitive state-of-the-art multi-task and transfer learning methods.

Keywords: Transfer Learning, Multi-Task Learning, Random Matrix Theory, Support Vector Machine, Classification.

1. Introduction

The methodology for a long time considered in machine learning has consisted in tackling each given (classification, regression, estimation) problem, hereafter referred to as a *task*, independently. This approach is in general counterproductive as it automatically discards a potentially rich source of data often available to perform more or less similar tasks. Multi Task Learning (MTL) precisely aims to handle this deficiency by connecting datasets and tasks so to improve the generalization performance of one or several specific target tasks. This framework has recently gained renewed interest (Yang et al., 2020; Caruana, 1997; Collobert and Weston, 2008), given the availability of gigantic datasets (such as huge prelabelled image databases) and costly trained learning machines (such as deep neural nets), which must be useful to help solve learning tasks involving much fewer labelled data. Beyond this resurgence, numerous applications inherently benefit from a MTL approach, of which we may cite a few examples: prediction of student test results for a collection of schools (Aitkin and Longford, 1986), patient survival estimates in different clinics (Harutyunyan et al., 2017; Caruana et al., 1996), values of possibly related financial indicators (Allenby and Rossi, 1998), preference modelling of many individuals in a marketing context (Greene, 2000), etc.

Carefully modelling the relatedness between tasks has long been claimed to be the most critical determinant of the MTL algorithm performance. Several such models have been considered in the literature: task relatedness can be modelled by assuming that the parameters relating the tasks lie on a low dimensional manifold (Argyriou et al., 2007; Agarwal et al., 2010); these relating parameters may alternatively be assumed to be close in norm (Evgeniou and Pontil, 2004; Xu et al., 2013) or be distributed according to similar priors (Xue et al., 2007; Yu et al., 2005). However, for all these models, a failure in properly matching the task parameters is often likely to induce possibly severe cases of *negative learning*, that is occurrences where additional tasks play *against* rather than in favor of the target task objective. These cases of negative learning are difficult to anticipate as few theoretical works are amenable to prepare the experimenter to these scenarios. In the present work, we adopt a similar strategy as in (Evgeniou and Pontil, 2004), but with a strong theoretical background which will automatically eliminate the risks of negative learning.

In detail, the article (Evgeniou and Pontil, 2004), the spirit of which is followed here, is inspired by the natural extension of support vector machines (SVMs) (Vapnik, 2005) to a multiple, say k, task setting, by paralleling k SVMs but constraining their parameters (specifically, the k separating hyperplane normal vectors $\omega_1, \ldots, \omega_k$) to be "close" to each other. This is enforced by simply imposing that $\omega_i = \omega_0 + v_i$ for some common hyperplane normal vector ω_0 and dedicated hyperplane normal vectors v_i. The norm of the vectors v_i is controlled through an additional hyperparameter λ to strengthen or relax task relatedness. This is the approach followed in the present article, to the noticeable exception that the fully explicit least-square SVM (LSSVM) (Xu et al., 2013) rather than a margin-based SVM is considered. In addition to only marginally altering the overall behavior of the MTL algorithm of (Evgeniou and Pontil, 2004), the LSSVM approach entails more explicit, more tractable, as well as more insightful results, let alone numerically cheaper implementations. As a matter of fact, by a now well-established universality argument of large dimensional

statistics, it has been shown in closely related works (Mai and Liao, 2019) that quadratic (least-square) cost functions are asymptotically optimal (as the data dimension and number increase) and uniformly outperform alternative costs (such as margin-based methods or logistic approaches), even in a classification setting; this argument further motivates to consider first and foremost the least square version of MTL-SVM.

The intricate nature of the MTL framework, even in its simplest MTL-SVM version (Evgeniou and Pontil, 2004), has so far left little room to sound and practical useful theoretical analysis – which we believe to have been a main reason for its decayed importance before the resurgence of the powerful deep learning tools, in capacity to tip the performance-complexity tradeoff. Among existing theoretical analyses of MTL, an "extended VC dimension" approach to retrieve bounds on the generalization performance is proposed in (Baxter, 2000; Ben-David and Schuller, 2003). Using Bayesian and information theoretic arguments, (Baxter, 1997) answers the question of the minimal information and number of samples per task required to learn k parallel tasks. However, these works only provide loose bounds and orders of magnitude which, if convenient to decide on the impossibility to reach a target objective, do not provide any satisfying accurate performance evaluations, nor do they allow for an optimal hyperparametrization of the MTL framework which, as we shall see, is of dramatic importance.

Following on a recent line of breakthroughs in applied random matrix theory, and specifically walking in the steps of (Liao and Couillet, 2019; Mai et al., 2019) which study a single-task LSSVM adapted to supervised (Liao and Couillet, 2019) and semi-supervised (Mai et al., 2019) learning, the article develops a theoretical framework to exhaustively study the behavior and maximize the performance of a k-task m-class MTL-LSSVM framework, under the regime of numerous (n) and large (p) data, i.e., $n, p \to \infty$ with $n/p \to c_0 \in (0, \infty)$. The data are here modelled as a mixture of km *concentrated random vectors*, i.e., for $\mathbf{x}$ a data of class j ($j \in \{1, \ldots, m\}$) for Task i ($i \in \{1, \ldots, k\}$), $\mathbf{x} \sim \mathscr{L}_{ij}(\mu_{ij}, \Sigma_{ij})$, where $\mathscr{L}_{ij}(\mu, \Sigma)$ is the law of a Lipschitz-concentrated random vector (Ledoux, 2001) with statistical mean $\mu \in \mathbb{R}^p$ and covariance $\Sigma \in \mathbb{R}^{p \times p}$. For instance, $\mathbf{x} = \varphi_{ij}(\mathbf{z})$ for $\mathbf{z} \sim \mathcal{N}(0, I_q)$, $\varphi_{ij} : \mathbb{R}^q \to \mathbb{R}^p$ a 1-Lipschitz function and $\lim q/p \in (0, \infty)$. The main results and practical consequences of the article may be summarized as follows:

1. under the regime of large dimensional datasets, the MTL-LSSVM algorithm has an asymptotically predictable behavior and thus a predictable performance; in particular, under the further assumptions of two classes per task ($m = 2$) and equal identity covariance of the mixture ($\Sigma_{ij} = I_p$ for all i, j), this behavior summarizes as a very insightful *small dimensional* (of size the number of tasks k and not the number n or dimension p of the data) functional (i) of all inner products $\Delta\mu_i^\mathsf{T} \Delta\mu_{i'}$, $i, i' \in \{1, \ldots, k\}$, where $\Delta\mu_i = \mu_{i1} - \mu_{i2}$, (ii) of the proportions $c_{ij} = \lim n_{ij}/n$ between the number of data n_{ij} of class j in Task i and the overall number n of data, and (iii) of the hyperparameters λ (task relatedness) and $\gamma_1, \ldots, \gamma_k$ (task-wise LSSVM regularization parameters) of the MTL problem;

2. a fundamental aspect of the (LS)SVM framework is to associate each training data $\mathbf{x}$ to a label $y \in \{-1, 1\}$; we demonstrate that this choice is a dangerous source of *negative transfer*; most importantly, we show that to each $\mathbf{x}$ must be associated an *"optimal"*

score[1] rather than a label y, which only depends on the class and task of $\mathbf{x}$; this optimal score is provided in explicit form by the large dimensional analysis; under this choice of optimal scores y, the performance of MTL-LSSVM is necessarily improved over parallel independent single-task LSSVMs, and *discards all risks of negative transfer*;

3. a further aspect of the MTL-(LS)SVM approach is that, in a two-class setting, for an unlabelled data $\mathbf{x}$ to be associated to class $j \in \{1, 2\}$ for Task i, a binary decision of the type $g_i(\mathbf{x}) \underset{\mathscr{C}_2}{\overset{\mathscr{C}_1}{\gtrless}} 0$ is performed; we show that this decision rule is in general biased, not only due to imbalances in the number of available data per class and per task, but also by the data statistics and the MTL hyperparameters (unless $\Sigma_{ij} = I_p$ for each i, j); similar to an optimal choice of the training data "labels", in the all-identity covariance setting ($\Sigma_{ij} = I_p$), we establish an optimal threshold ζ_i which minimizes the probability of misclassification: ζ_i can be consistently estimated and thus used in practice;

4. the assumption of a mixture of concentrated random vectors for the data samples is far from anecdotal: concentrated random vectors form a broad and rich family of random vectors, which can mimic extremely realistic data, as is the case of the output of generative adversarial networks (GANs) proved to be, by definition, concentrated random vectors (Seddik et al., 2020); the article proves a universality result: the asymptotic performance (as $p, n \to \infty$) of MTL-LSSVM only depends on the statistics μ_{ij} and Σ_{ij} of the mixture model, thereby behaving *as if* the data followed a mere Gaussian mixture model; this strongly suggests that the proposed improved algorithm and its performances are applicable to a wide range of real data;

5. a series of concrete applications, to hypothesis testing using external tasks, to transfer learning, and to multi-class classification are provided, optimized and confronted to competing methods; these applications have the strong advantage to have predictable performances: this is particularly crucial to appropriately set decision thresholds for type I and II errors in hypothesis testing, as well as to predict *before running the algorithms* their anticipated performances;

6. a simulation campaign on real datasets is performed which (i) confirms, as strongly suggested by Item 4, the strong adequacy between the empirical and theoretical results and (ii) demonstrates the large superiority of the proposed algorithm over competing methods.

In a nutshell, by exploiting recent advances in applied random matrix theory, the article provides a modern vision to multi-task and transfer learning. This vision is here turned into an elementary but cost-efficient algorithm, which relies on base principles, but which both largely outperforms competing (sometimes complex) methods and provides strong theoretical guarantees. As a side note, we must insist that our present objective is to study and improve "data-generic" multi-task learning mechanisms under no structural assumption on the data; this is quite unlike recent works exploiting convolutive techniques in deep neural

1. This notion of optimality will be properly defined in the article.

nets to perform transfer or multi-task learning mostly for computer vision-oriented tasks, as in e.g., (Zhuang et al., 2020; Krishna and Kalluri, 2019).

In order to best capture the main intuitions drawn from the large dimensional analysis, after a rigorous introduction of the multitask learning framework in Section 2, a first highlight of our main contributions under the qualitatively more telling setting of binary tasks ($m = 2$) with data of equal identity covariance ($\Sigma_{ij} = I_p$) is proposed in Section 3. The technical details under the most generic data modelling setting as well as the most general technical result are then provided in Section 4. A broad series of applications is provided in Section 5. Extensive simulations are then proposed in Section 6, which corroborate our theoretical findings and show their resilience and compatibility to real data settings.

Reproducibility. Matlab codes of the main algorithms and results provided in the article are available at `https://github.com/maliktiomoko/RMT-MTLLSSVM.git`.

Notation. The following notations and conventions will be used throughout the article: $\mathbb{1}_n \in \mathbb{R}^n$ is the vector of all ones, $e_m^{[n]} \in \mathbb{R}^n$ is the canonical vector of $\mathbb{R}^n$ with $[e_m^{[n]}]_i = \delta_{mi}$, and $e_{ij}^{[2k]} \equiv e_{2(i-1)+j}^{[2k]}$. Similarly, $E_{ij}^{[n]} \in \mathbb{R}^{n\times n}$ is the canonical matrix of $\mathbb{R}^{n\times n}$ with $[E_{ij}^{[n]}]_{ab} = \delta_{ia}\delta_{jb}$. The notation $A \otimes B$ for matrices or vectors A, B is the Kronecker product. The notation $A \odot B$ for matrices or vectors A, B is the Hadamard product. $\mathscr{D}_x$ stands for a diagonal matrix containing on its diagonal the elements of the vector x and $A_{i\cdot}$ is the i-th row of matrix A.

2. The Multi Task Learning Framework

2.1 The deterministic 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", each task consisting of an m-class classification problem (m being the same for each task). Specifically, letting $X = [X_1, \ldots, X_k]$, Task i is a classification problem from the training samples $X_i = [X_i^{(1)}, \ldots, X_i^{(m)}] \in \mathbb{R}^{p\times n_i}$ 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 $\mathscr{C}_j$, $j \in \{1, \ldots, m\}$, for Task i. In particular, $n = \sum_{i=1}^k n_i$ and $n_i = \sum_{j=1}^m n_{ij}$ for each $i \in \{1, \ldots, k\}$.

To each datum $x_{il}^{(j)} \in \mathbb{R}^p$ of the training set is attached a corresponding output vector (or score) $y_{il}^{(j)} \in \mathbb{R}^m$. Correspondingly to the notation X, X_i and $X_i^{(j)}$, let $Y = [Y_1^\mathsf{T}, \ldots, Y_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{n\times m}$ be the matrix of the m-dimensional outputs of all data, where $Y_i = [Y_i^{(1)\mathsf{T}}, \ldots, Y_i^{(m)\mathsf{T}}]^\mathsf{T} \in \mathbb{R}^{n_i\times m}$ and $Y_i^{(j)} = [y_{i1}^{(j)}, \ldots, y_{in_{ij}}^{(j)}]^\mathsf{T} \in \mathbb{R}^{n_{ij}\times m}$ the matrix of all outputs for Task i.

In the standard MTL learning approach (Evgeniou and Pontil, 2004; Xu et al., 2013), one would naturally set $y_{il}^{(j)} = e_j^{[m]}$, i.e., all data of class $\mathscr{C}_j$ are affected a hot-bit in position j. As claimed in the introduction and as we shall see, this hot-bit allocation approach is at the source of deleterious performances, such as negative transfer effects, and we thus voluntarily do not enforce any constraint on the vector $y_{il}^{(j)}$ at this point.

Before inserting the data-score pairs (X, Y) into the MTL-LSSVM framework, it is convenient to "center" the data X to eliminate additional sources of bias. This centering

operation could be performed either on the whole dataset X, or task-wise on each X_i, or even class-wise on each $X_i^{(j)}$. In (Evgeniou and Pontil, 2004; Xu et al., 2013) this centering operation is not performed (which essentially boils down to centering X itself). We choose here to center the data task-wise, and this, for two reasons: (i) centering the whole dataset induces dependencies across tasks so that, even by enforcing the hyperplane controlling factor λ to decorrelate the tasks (i.e., $\lambda \to \infty$; see next), residual dependence must remain and negative transfer can still appear, (ii) class-wise centering has the double deleterious effect of cancelling an important discrimination factor of the classes (i.e., their difference in statistical mean) and of necessitating a complex treatment to classify new (unlabelled) input data. Inappropriate centering choices would induce biases and undesired residual terms in our theoretical derivation, which further justifies our present task-wise centering choice (see e.g., Remark 3). Specifically, the MTL-LSSVM algorithm studied here is based, not on the data X_i but on their centered version

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

and we will systematically consider the data-score pair $(\mathring{X}, Y)$, where $\mathring{X} = [\mathring{X}_1, \ldots, \mathring{X}_k]$ rather than (X, Y).

Having pre-treated the input data, we are in position to introduce the MTL-LSSVM framework. The MTL-LSSVM algorithm aims to predict, relative to each task i, an output score vector $\mathbf{y}_i \in \mathbb{R}^m$ for any new input vector $\mathbf{x} \in \mathbb{R}^p$. To this end, MTL-LSSVM determines k "hyperplane normal-vector" matrices $W = [W_1, W_2, \ldots, W_k] \in \mathbb{R}^{p \times km}$ which take the form $W_i = W_0 + V_i$ for some common W_0 and individual task-wise matrices $V = [V_1, \ldots, V_k]$ and biases $b = [b_1^\mathsf{T}, b_2^\mathsf{T}, \ldots, b_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{k \times m}$. These parameters are set to minimize the objective function

$$\min_{(W_0, V, b) \in \mathbb{R}^{p \times m} \times \mathbb{R}^{p \times km} \times \mathbb{R}^{k \times m}} \mathscr{J}(W_0, V, b) \tag{1}$$

where

$$\mathscr{J}(W_0, V, b) \equiv \frac{1}{2\lambda} \mathrm{tr} \left(W_0^\mathsf{T} W_0 \right) + \frac{1}{2} \sum_{i=1}^k \frac{\mathrm{tr} \left(V_i^\mathsf{T} V_i \right)}{\gamma_i} + \frac{1}{2} \sum_{i=1}^k \mathrm{tr} \left(\xi_i^\mathsf{T} \xi_i \right)$$

$$\xi_i = Y_i - \left(\frac{\mathring{X}_i^\mathsf{T} W_i}{kp} + \mathbb{1}_{n_i} b_i^\mathsf{T} \right), \quad \forall i \in \{1, \ldots, k\}.$$

This is a classical LSSVM formulation in which the quadratic cost $\mathrm{tr}(\xi_i^\mathsf{T} \xi_i)$ replaces the boundary constraint of margin-based SVM and where the costs $\mathrm{tr}(W_0^\mathsf{T} W_0)$ and $\mathrm{tr}(V_i^\mathsf{T} V_i)$ are reminiscent of the hyperplane normal-vector norm minimization of classical SVM.

What is specific to the MTL approach is first the hyperparameter λ which enforces or relaxes the relatedness between tasks and the introduction of k extra parameters $\gamma_1, \ldots, \gamma_k$ which enforce a correct classification of the data in their respective classes. Similarly to (Evgeniou and Pontil, 2004), we place the hyperparameters γ_i as a prefactor of $\mathrm{tr}(V_i^\mathsf{T} V_i)$, rather than as a prefactor of $\mathrm{tr}(\xi_i^\mathsf{T} \xi_i)$; this differs from the normalization scheme proposed in (Xu et al., 2013). This choice is more flexible in the following sense: for a fixed value of λ,

increasing all ratios $\frac{\lambda}{\gamma_i}$ "blurs" the difference between tasks and thus turns the optimization scheme into a single-task SVM (because the optimal V_i's need then be set to zero in the limit); for fixed values of the γ_i's instead, small ratios $\frac{\lambda}{\gamma_i}$ decorrelate the tasks (the optimal W_0 being close to zero). Note however that, unlike in (Evgeniou and Pontil, 2004), we choose to use here one hyperparameter γ_i per task instead of a common one. As will be seen next, this choice is more meaningful and of course offers more flexibility.

In passing, remark that the linear common-hyperplane condition $W_i = W_0 + V_i$, imposes by definition that all V_i's be of the same size $\mathbb{R}^{p\times m}$: this severely constrains (i) the data in each task to be of the same dimension p and (ii) the number of classes per task to be the same (m). Further linear or even non-linear relaxation schemes for W_i of the type $W_i = V_i + f_i(W_0)$ for some operator f_i could be envisioned to relax this constraint. This however goes beyond the scope of the present article, which seeks to provide insights and optimality into a simplified (yet already non-trivial) form of MTL-LSSVM.

As for the choice of the hyperparameters $\lambda, \gamma_1, \ldots, \gamma_k$, as well as of the score matrix Y which we recall was left open, it is treated independently and is dictated, not by the present optimization scheme, but by a ultimate objective, such as minimizing the misclassification rate for a specific target class. These more applied considerations will be made in Section 5.

Remark 1 (LSSVM classification versus regression) *It may be disputed that the optimization framework* (1) *takes a regression rather than a classification form. It appears that, under a binary-class LSSVM framework with scores $y_i \in \{\pm 1\}$, the classification constraint (of the form $y_i(W^T x_i + b_i) - 1 = \xi_i$) or the regression constraint (of the form $y_i - W^T x_i + b_i = \xi_i$) are associated to the same losses, thereby leading to the same classification solution and performance. Yet, as will become clear in the following, in addition for the solution of* (1) *to be explicit and theoretically tractable (which is not the case of alternative schemes such as margin-based SVM, logistic regression, Adaboost, etc.), the aforementioned flexibility in the score matrix Y largely outbalances the "failure" of treating a classification problem by means of a regression optimization scheme. Besides, under the large dimensional theoretical framework presently studied, recent works in related problems (Mai and Liao, 2019) forcefully suggest that the square loss is optimal to deal with large dimensional data as it uniformly outperforms all alternative cost functions.*

Being a quadratic cost optimization under linear constraints, (1) is easily solved using its dual formulation by introducing Lagrangian parameters $\alpha_i \in \mathbb{R}^{n_i \times m}$ for each task i (see details in Section A.1). The solution is explicit and is as follows.

Proposition 2 *The solution to* (1) *is given by*

$$
\begin{aligned}
W_0 &= \left(\mathbb{1}_k^\mathsf{T} \otimes \lambda I_p \right) Z\alpha \\
W_i &= \left({e_i^{[k]}}^\mathsf{T} \otimes I_p \right) AZ\alpha \\
b &= (P^\mathsf{T} Q P)^{-1} P^\mathsf{T} Q Y
\end{aligned}
$$

where

$$
\begin{aligned}
Z &= \begin{pmatrix} \mathring{X}_1 & & \\ & \ddots & \\ & & \mathring{X}_k \end{pmatrix} \in \mathbb{R}^{kp \times n} \\
A &= \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T} \right) \otimes I_p \in \mathbb{R}^{kp \times kp} \\
\alpha &= Q(Y - Pb), \quad Q = \left(\frac{1}{kp} Z^\mathsf{T} A Z + I_n \right)^{-1} \in \mathbb{R}^{n \times n} \\
P &= \begin{pmatrix} \mathbb{1}_{n_1} & & \\ & \ddots & \\ & & \mathbb{1}_{n_k} \end{pmatrix} \in \mathbb{R}^{n \times k}.
\end{aligned}
$$

Despite the apparent intricate expression of W_i, it must be stressed that W_i "essentially" takes the form of the standard solution to a ridge regression (or regularized least-square) problem as the term $AZQY$ (in which $Q = (\frac{1}{kp} Z^\mathsf{T} A Z + I_n)^{-1}$) appearing in the expended form of W_i confirms. From a technical standpoint, the large dimensional statistical behavior of the matrix Q, known as the *resolvent* of $\frac{1}{kp} Z^\mathsf{T} A Z$ in random matrix theory, plays a central role in the analysis. More specific to the MTL framework, note the interesting isolation of the data subsets $\mathring{X}_i$ in the data matrix Z (it is not possible, to the best of our knowledge, to "linearly" express W_i as a function of $\mathring{X}$ itself); the elements $\mathring{X}_i$ are then "mixed" by the term $\lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}$ appearing in matrix A, from which it naturally comes that, in the limit $\lambda \to 0$, MTL-LSSVM boils down to k independent LSSVMs with $\mathscr{D}_\gamma$ imposing weights $\gamma_1, \ldots, \gamma_k$ on each data subset.

From Proposition 2, for any new data point $\mathbf{x} \in \mathbb{R}^p$, the classification score vector $g_i(\mathbf{x}) \in \mathbb{R}^m$ for Task i, is then defined by

$$
g_i(\mathbf{x}) = \frac{1}{kp} W_i^\mathsf{T} \mathring{\mathbf{x}} + b_i = \frac{1}{kp} \alpha^\mathsf{T} Z^\mathsf{T} A \left(e_i^{[k]} \otimes \mathring{\mathbf{x}} \right) + b_i \tag{2}
$$

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.

This formulation, along with the next remark, confirm again the relevance of a task-wise, rather than class-wise, centering of the data X, which allows for a well-defined expression of $\mathring{\mathbf{x}}$.

Remark 3 (Shift invariance of the scores) *If the columns of $Y_i \in \mathbb{R}^{n_i \times m}$ are shifted by some constant vector $P\bar{\mathscr{Y}}$ for some (small dimensional) matrix $\bar{\mathscr{Y}} \in \mathbb{R}^{k \times m}$, i.e., if all data of the same task are affected by the same shift of their scores (or labels), then we find that the Lagrangian parameter α^{shift} after the shift is*

$$
\alpha^{\text{shift}} = Q \left(I_n - P(P^\mathsf{T} Q P)^{-1} P^\mathsf{T} Q \right) (Y + P\bar{\mathscr{Y}}) = \alpha.
$$

As such, the matrix $W_i = (e_i^{[k]\mathsf{T}} \otimes I_p) AZ\alpha$ and, consequently, the performance of MTL-LSSVM are insensitive to a simultaneous shift of all the scores of each task.

2.2 Statistical modelling and the large dimensional setting

In order to draw insights into the behavior of MTL-LSSVM and evaluate its performance, the article proposes to first model the dataset X as a mixture of concentrated random vectors and then to assume the dimensions p, n of X to be sufficiently large for deterministic (and predictable) concentration behavior to occur.

Assumption 1 (Distribution of X and x) *There exist two constants $C, c > 0$ (independent of n, p) such that, for any 1-Lipschitz function $f : \mathbb{R}^{p\times n} \to \mathbb{R}$,*

$$\forall t > 0,\ \mathbb{P}(|f(X) - m_{f(X)}| \geq t) \leq Ce^{-(t/c)^2}$$

where m_Z is a median of the random variable Z. We further impose that the columns of X be independent and that the $x_{il}^{(j)}$, for $l \in \{1, \ldots, n_{ij}\}$, be distributed according to the same law $\mathscr{L}_{ij}$. These conditions guarantee the existence of a mean and covariance for the columns of X and we denote, for all $l \in \{1, \ldots, n_{ij}\}$,

$$\mu_{ij} \equiv \mathbb{E}[x_{il}^{(j)}]$$
$$\Sigma_{ij} \equiv \mathrm{Cov}[x_{il}^{(j)}].$$

Furthermore, the dummy variable $\mathbf{x} \in \mathbb{R}^p$ used for testing is independent of X, and distributed according to one of the laws $\mathscr{L}_{ij}$.

Assumption 1 notably encompasses the following scenarios: the $x_{il}^{(j)}$'s are (i) independent Gaussian random vectors $\mathcal{N}(\mu_{ij}, \Sigma_{ij})$, (ii) independent random vectors uniformly distributed on the $\mathbb{R}^p$ sphere of radius $\sqrt{p}$ and, most importantly, (iii) any 1-Lipschitz transformation $\varphi_{ij}(z_{il}^{(j)})$ with $z_{il}^{(j)}$ itself a concentrated random vector. Scenario (iii) is particularly relevant to model very realistic data by means of advanced non-linear generative models, as recently demonstrated in (Seddik et al., 2019) in the specific example of generative adversarial networks (GANs). As such, Assumption 1 offers the flexibility to assume either synthetic Gaussian mixture models, or very realistic and advanced generative data models. A core result of the present article consists in showing that, for n, p large, either scenario leads to the same asymptotic performance for MTL-LSSVM (which thus only depends on the statistical means and covariances of the data).

Since all data $x_{il}^{(j)}$, $l \in \{1, \ldots, n_{ij}\}$, are identically distributed, we will further impose that their associated scores $y_{il}^{(j)} \in \mathbb{R}^m$ be identical. That is, $y_{i1}^{(j)} = \ldots = y_{in_{ij}}^{(j)} \equiv \mathscr{Y}_{ij}$ within every class j of each task i. The score matrix $Y \in \mathbb{R}^{n\times k}$ may then be reduced under the form

$$Y = \left[\mathscr{Y}_{11}\mathbb{1}_{n_{11}}^\mathsf{T}, \ldots, \mathscr{Y}_{km}\mathbb{1}_{n_{km}}^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^{n\times m}$$

for $\mathscr{Y} = [\mathscr{Y}_{11}, \ldots, \mathscr{Y}_{km}]^\mathsf{T} \in \mathbb{R}^{km\times m}$. From Remark 3, it is also clear that, the performances of MTL-LSSVM being insensitive to a constant shift in the scores $\mathscr{Y}_{i1}, \ldots, \mathscr{Y}_{im}$ in every given task i, the centered version $\mathring{\mathscr{Y}} = [\mathring{\mathscr{Y}}_{11}, \ldots, \mathring{\mathscr{Y}}_{km}]^\mathsf{T}$ of $\mathscr{Y}$, where

$$\mathring{\mathscr{Y}}_{ij} \equiv \mathscr{Y}_{ij} - \sum_{j=1}^{m} \frac{n_{ij}}{n_i}\mathscr{Y}_{ij},$$

will naturally appear at the core of the upcoming results.

Although practical data will of course be considered to be of finite dimension p and number n, it will indeed be convenient, for technical reasons, to work under the following large dimensional random matrix assumption.

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

With these notations and assumptions in place, we are in position to present the main results of the article. Yet, before entering the technical details of the large dimensional analysis of the performance of the MTL-LSSVM framework, the next section first provides a highlight of the main contributions and intuitions drawn by the analysis. To this end, it is convenient to temporarily restrict the setting to binary classes ($m = 2$) and to an isotropic mixture model for the data X, i.e., $\Sigma_{ij} = I_p$ for each measure $\mathscr{L}_{ij}$. The most general and slightly more technical setting ($m \geq 2$ and non-isotropic mixture data modelling) is considered in full in Section 4.

3. Highlights of the main results

To simplify the exposition of our main results, without impacting their core conclusions, in this section, Assumptions 1–2 are further restricted to the binary-classification setting ($m = 2$) and to measures $\mathscr{L}_{ij}$ of equal covariance $\Sigma_{ij} = I_p$, for all i, j.

The advantage of the isotropic ($\Sigma_{ij} = I_p$) condition is that all asymptotic results can be expressed under the form of low-dimensional matrix formulations (of size scaling with k but not with p, n). Adjoined to the $m = 2$ assumption, the isotropic model further guarantees a simplified form for (i) the (asymptotically) optimal labels Y, (ii) the optimal decision thresholds ζ_i, and (iii) the asymptotic performances of MTL-LSSVM, all of which can be estimated consistently as $p, n \to \infty$. Consequently, this simplified setting has the strong benefit to give rise to a first cost-efficient and robust multitask classification algorithm (Algorithm 1) which, for practical data, makes the approximation that $\Sigma_{ij} \propto I_p$.

The binary setting does not a priori alter any of the previously introduced notations which stand with $m = 2$. Yet, it is particularly convenient in this setting to recast the score vectors $y_{il}^{(j)} \in \mathbb{R}^m$ into scalar scores $y_{il}^{\mathrm{bin}(j)} \in \mathbb{R}$. In a standard classification context, this would correspond to turning a two-dimensional hot-bit vector $e_j^{[2]}$ into a signed scalar ± 1; as we recall that $y_{il}^{(j)}$ is here considered as a *real* score (rather than a binary label) vector, to us this is equivalent to turning a score vector into a scalar score. Matrix $Y \in \mathbb{R}^{n \times m}$ similarly now becomes a score vector $y^{\mathrm{bin}} \in \mathbb{R}^n$, and in particular we define $\mathring{y}^{\mathrm{bin}} = [\mathring{y}_{11}^{\mathrm{bin}}, \ldots, \mathring{y}_{k2}^{\mathrm{bin}}]^\mathsf{T} \in \mathbb{R}^{2k}$ with

$$\mathring{y}_{ij}^{\mathrm{bin}} \equiv y_{ij}^{\mathrm{bin}} - \left(\frac{n_{i1}}{n_i} y_{i1}^{\mathrm{bin}} + \frac{n_{i2}}{n_i} y_{i2}^{\mathrm{bin}} \right) \in \mathbb{R}$$

where $y_{ij}^{\mathrm{bin}} \equiv y_{i1}^{\mathrm{bin}(j)} = \ldots = y_{in_{ij}}^{\mathrm{bin}\,(j)} \in \mathbb{R}$ is the common score assigned to the identically distributed data of class j for Task i. Correspondingly, the sought-for (W_i, b_i) collection of m hyperplanes of (1) becomes a single hyperplane $(w_i^{\mathrm{bin}}, b_i^{\mathrm{bin}})$ with $w_i^{\mathrm{bin}} \in \mathbb{R}^p$ and $b_i^{\mathrm{bin}} \in \mathbb{R}$.

Yet, our present interest is only on the resulting score vector $g_i(\mathbf{x})$ which, replacing Y by y^{bin} in its expression (Equation 2), becomes the scalar test score

$$g_i^{\text{bin}}(\mathbf{x}) \equiv \frac{1}{kp}(y^{\text{bin}} - Pb)^\mathsf{T} QZ^\mathsf{T} A \left(e_i^{[k]} \otimes \mathring{\mathbf{x}}\right) + [(P^\mathsf{T} QP)^{-1} P^\mathsf{T} Q y^{\text{bin}}]_i \in \mathbb{R}.$$

3.1 Theoretical analysis and large dimensional intuitions

Under the isotropic and binary-class setting, as $n, p \to \infty$ according to Assumption 2, the theoretical performance of MTL-LSSVM explicitly depends on two fundamental and isolated quantities: the data-related matrix $\mathscr{M} \in \mathbb{R}^{2k \times 2k}$ and the hyperparameter matrix $\mathscr{A} \in \mathbb{R}^{k \times k}$:

$$\begin{aligned}
\mathscr{M} &= \sum_{i,i'=1}^{k} \Delta\mu_i^\mathsf{T} \Delta\mu_{i'} \left(E_{ii'}^{[k]} \otimes \mathbb{c}_i \mathbb{c}_{i'}^\mathsf{T}\right) \\
\mathscr{A} &= \left(I_k + \mathscr{D}_{\boldsymbol{\delta}^{[k]}}^{-\frac{1}{2}} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{-1} \mathscr{D}_{\boldsymbol{\delta}^{[k]}}^{-\frac{1}{2}}\right)^{-1}
\end{aligned}$$

where we introduced the shortcut notations

$$\Delta\mu_i \equiv \mu_{i1} - \mu_{i2}, \quad \mathbb{c}_i \equiv \sqrt{c_{i1}/c_i}\sqrt{c_{i2}/c_i} \begin{bmatrix} \sqrt{c_{i2}/c_i} \\ -\sqrt{c_{i1}/c_i} \end{bmatrix}$$

and where $\boldsymbol{\delta}^{[k]} = [\boldsymbol{\delta}_1^{[k]}, \ldots, \boldsymbol{\delta}_k^{[k]}]^\mathsf{T}$ are the unique positive solutions to the implicit system of k equations

$$\boldsymbol{\delta}_i^{[k]} = \frac{c_i}{c_0} - \mathscr{A}_{ii}, \quad i \in \{1, \ldots, k\}. \tag{3}$$

In anticipation of future needs, it is convenient to further introduce the $2k$-dimensional variant $\boldsymbol{\delta}^{[2k]} = [\boldsymbol{\delta}_{11}^{[2k]}, \ldots, \boldsymbol{\delta}_{k2}^{[2k]}]^\mathsf{T} \in \mathbb{R}^{2k}$ where

$$\boldsymbol{\delta}_{ij}^{[2k]} = c_0 \frac{c_{ij}}{c_i} \boldsymbol{\delta}_i^{[k]}. \tag{4}$$

The asymptotic performances of MTL-LSSVM will be shown to solely depend on X through the matrices $\mathscr{M}$ and $\mathscr{A}$, which thus play the role of (asymptotically) *sufficient statistics*. It is particularly important to stress that, despite the quite generic concentration assumption on X (Assumption 1), when $\Sigma_{ij} = I_p$, only the k^2 inner products $\Delta\mu_i^\mathsf{T}\Delta\mu_{i'}$ and the $2k$ *class-wise* dimensionality ratios c_{ij}/c_i intervene in the expression of $\mathscr{M}$ – so in particular none of the higher order moments of X are accounted for, nor the absolute task-wise dimension ratios c_i. As for $\mathscr{A}$, it captures instead the information about the impact of the hyperparameters $\lambda, \gamma_1, \ldots, \gamma_k$ as well as the *task-wise* dimensionality ratios $c_1, \ldots, c_k$ and the data number-to-dimension ratio c_0. In the expression of the MTL-LSSVM performance, these two matrices combine into the core matrix $\Gamma \in \mathbb{R}^{2k \times 2k}$

$$\Gamma = \left(I_{2k} + \left(\mathscr{A} \otimes \mathbb{1}_2 \mathbb{1}_2^\mathsf{T}\right) \odot \mathscr{M}\right)^{-1} \tag{5}$$

where we recall that '$\odot$' is the Hadamard (element-wise) matrix product.

Theorem 4 (Asymptotics of $g_i^{\mathrm{bin}}(\mathbf{x})$) *Under Assumptions 1–2, with $m = 2$ and $\Sigma_{ij} = I_p$, for a test data $\mathbf{x}$ with $\mathbb{E}[\mathbf{x}] = \mu_{ij}$ and $\mathrm{Cov}[\mathbf{x}] = I_p$, as $p, n \to \infty$,*

$$g_i^{\mathrm{bin}}(\mathbf{x}) - G_{ij} \xrightarrow{\text{a.s.}} 0, \quad G_{ij} \sim \mathcal{N}(m_{ij}, \sigma_i^2)$$

in distribution, where, letting $\boldsymbol{m} = [m_{11}, \ldots, m_{k2}]^\mathsf{T}$ and the normalized forms $\boldsymbol{y}^{\mathrm{bin}} \equiv \mathcal{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}} y^{\mathrm{bin}}$, $\mathring{\boldsymbol{y}}^{\mathrm{bin}} = \mathcal{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}} \mathring{y}^{\mathrm{bin}}$, $\boldsymbol{m} = \mathcal{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}} m$, and $\boldsymbol{\sigma}_i^2 = \boldsymbol{\delta}_i^{[k]} \sigma_i^2$,

$$\begin{aligned} \boldsymbol{m} &= \boldsymbol{y}^{\mathrm{bin}} - \Gamma \mathring{\boldsymbol{y}}^{\mathrm{bin}} \\ \boldsymbol{\sigma}_i^2 &= (\mathring{\boldsymbol{y}}^{\mathrm{bin}})^\mathsf{T} \Gamma \mathscr{V}_i \Gamma \mathring{\boldsymbol{y}}^{\mathrm{bin}} \end{aligned}$$

with

$$\begin{aligned} \mathscr{V}_i &= \mathscr{D}_{\mathscr{K}_{i\cdot}^\mathsf{T} \otimes \mathbb{1}_2} + \left(\mathscr{A} \mathscr{D}_{\mathscr{K}_{i\cdot}^\mathsf{T} + e_i^{[k]} \mathscr{A}} \otimes \mathbb{1}_2 \mathbb{1}_2^\mathsf{T} \right) \odot \mathscr{M} \\ \mathscr{K} &= \frac{c_0}{k} [\mathscr{A} \odot \mathscr{A}] \left(\mathscr{D}_c - \frac{c_0}{k} [\mathscr{A} \odot \mathscr{A}] \right)^{-1}. \end{aligned}$$

Theorem 4 interestingly indicates that the (asymptotic) statistics of the classification scores $g_i^{\mathrm{bin}}(\mathbf{x})$, for $1 \leq i \leq k$, reduce to a mere functional of $2k$-*dimensional deterministic vectors and matrices*. In particular, $g_i^{\mathrm{bin}}(\mathbf{x})$ depends on the data statistical means $\mu_{i'j'}$, $1 \leq i' \leq k$, $1 \leq j' \leq 2$, and on the hyperparameters λ and $\gamma_1, \ldots, \gamma_k$ mostly through the $2k$-dimensional matrix Γ (and more marginally through $\mathscr{V}_i$ and $\mathscr{K}$ for the variances).

Another non-trivial point to note is that, being in general non-diagonal, Γ acts on the centered scores (labels) $\mathring{y}_{i'j'}^{\mathrm{bin}}$ of all classes j' and tasks i' which, therefore, all influence the performances. It can thus be anticipated that, for the decision on a particular Task i to be successful, not only the scores y_{i1}^{bin} and y_{i2}^{bin}, but in fact all scores $y_{i'j'}^{\mathrm{bin}}$ across all classes and tasks, must be appropriately tuned.

Remark also that, in this isotropic ($\Sigma_{i'j'} = I_p$) setting, the variance σ_i^2 of the score $g_i^{\mathrm{bin}}(\mathbf{x})$ with $\mathbb{E}[\mathbf{x}] = \mu_{ij}$ only depends on i, and not on j. This is particularly convenient, as shown next, to devise an optimal decision rule for classification into class 1 or 2 for Task i.

From a more technical standpoint, comparing the exact expression of $g_i(\mathbf{x})$ in (2) and that of m_{ij} (i.e., the large dimensional approximation of $\mathbb{E}[g_i(\mathbf{x})]$), we may interpret the matrix $\Gamma \in \mathbb{R}^{2k \times 2k}$ as a "condensed" form of $Q \in \mathbb{R}^{n \times n}$. From the expression $(I_{2k} + \mathscr{A} \otimes \mathbb{1}_2 \mathbb{1}_2^\mathsf{T}) \odot \mathscr{M}$, observe that: (i) if $\lambda \ll 1$, then $\mathscr{A}$ is diagonal dominant and thus "filters out" in the Hadamard product all off-diagonal entries of $\mathscr{M}$ – that is, all the 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 $\lambda \sim 1$, then $\mathscr{A}$ 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 and the $\boldsymbol{\delta}_i^{[k]}$) of the entries of $\mathscr{M}$; specifically, letting $\gamma^{-1} = (\gamma_1^{-1}, \ldots, \gamma_k^{-1})^\mathsf{T}$, 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, letting aside the regularization effect of the $\boldsymbol{\delta}_i^{[k]}$'s, the off-diagonal $\Delta\mu_i^\mathsf{T} \Delta\mu_j$ term intervening in the expression of $\mathscr{M}$ is weighted by a coefficient $(\gamma_i \gamma_j)^{-1}$: the impact of the

$\gamma_{i'}$'s is thus strongly associated to the relevance of the correlation between tasks, and not only to the individual performances of the k isolated LSSVM tasks.

Section 4 provides a more general version (Theorem 8) of Theorem 4 for $m \geq 2$ classes per task and generic Σ_{ij}. The technical derivation of these two results, of limited interest at this point of the article, is also deferred to Section 4.

3.2 Decision threshold and label optimization

Since $g_i^{\text{bin}}(\mathbf{x})$ has a Gaussian limit centered about m_{ij} and with equal variance for $j = 1$ and $j = 2$, the (asymptotically) optimal decision for $\mathbf{x}$ to be allocated to class $\mathscr{C}_1$ or class $\mathscr{C}_2$ for Task i, i.e., the decision minimizing the averaged error probability under the prior $\mathbb{P}(\mathbf{x} \in \mathscr{C}_1) = \mathbb{P}(\mathbf{x} \in \mathscr{C}_2)$, is obtained by the "averaged-mean" test

$$g_i^{\text{bin}}(\mathbf{x}) \underset{\mathscr{C}_2}{\overset{\mathscr{C}_1}{\gtrless}} \zeta_i \equiv \frac{1}{2}(m_{i1} + m_{i2}) \tag{6}$$

the associated misclassification rate being

$$\begin{aligned} \epsilon_{i1} &\equiv \mathbb{P}\left(g_i^{\text{bin}}(\mathbf{x}) \geq \frac{m_{i1} + m_{i2}}{2} \middle| \mathbf{x} \in \mathscr{C}_1 \right) \\ &= \mathscr{Q}\left(\frac{m_{i1} - m_{i2}}{2\sigma_i} \right) + o(1) \end{aligned} \tag{7}$$

with m_{ij}, σ_i as in Theorem 4 and $\mathscr{Q}(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-\frac{u^2}{2}} du$.

It is of utmost interest at this point to recall that the asymptotics of $g_i^{\text{bin}}(\mathbf{x})$ from Theorem 4 (as from the more generic Theorem 8) depend in an elegant and simple manner on the training data scores $\mathscr{y}^{\text{bin}} = \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{-\frac{1}{2}} \boldsymbol{\mathscr{y}}^{\text{bin}}$. Using again the independence of σ_i^2 on the genuine class of $\mathbf{x}$, the vector $\boldsymbol{\mathscr{y}}^{\text{bin}\star}$ minimizing the misclassification rate for Task i simply reads:

$$\begin{aligned} \boldsymbol{\mathscr{y}}^{\text{bin}\star} &= \underset{\boldsymbol{\mathscr{y}}^{\text{bin}} \in \mathbb{R}^{2k}}{\arg\max} \frac{(m_{i1} - m_{i2})^2}{\sigma_i^2} \\ &= \underset{\boldsymbol{\mathscr{y}}^{\text{bin}} \in \mathbb{R}^{2k}}{\arg\max} \frac{\| (\boldsymbol{\mathscr{y}}^{\text{bin}})^\mathsf{T} (I_{2k} - \Gamma) \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{-\frac{1}{2}} (e_{i1}^{[2k]} - e_{i2}^{[2k]}) \|^2}{(\boldsymbol{\mathscr{y}}^{\text{bin}})^\mathsf{T} \Gamma \mathscr{V}_i \Gamma \boldsymbol{\mathscr{y}}^{\text{bin}}} \end{aligned}$$

for which the solution is explicitly defined, *up to an arbitrarily multiplicative constant (as it maximizes a ratio) and up to an arbitrary additive constant (as per Remark 3)*, by:

$$\boldsymbol{\mathscr{y}}^{\text{bin}\star} = \Gamma^{-1} \mathscr{V}_i^{-1} [(\mathscr{A} \otimes \mathbb{1}_2 \mathbb{1}_2^\mathsf{T}) \odot \mathscr{M}] \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{-\frac{1}{2}} (e_{i1}^{[2k]} - e_{i2}^{[2k]}). \tag{8}$$

and, for this choice of $\boldsymbol{\mathscr{y}}^{\text{bin}\star}$, the corresponding (asymptotically) optimal classification error ϵ_{i1} defined in (7) is then

$$\epsilon_{i1}^\star = \mathscr{Q}\left(\frac{1}{2} \sqrt{(e_{i1}^{[2k]} - e_{i2}^{[2k]})^\mathsf{T} \mathscr{G} (e_{i1}^{[2k]} - e_{i2}^{[2k]})} \right) \tag{9}$$

for $\mathscr{G} = \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}}[(\mathscr{A} \otimes \mathbb{1}_2\mathbb{1}_2^\mathsf{T}) \odot \mathscr{M}]\mathscr{V}_i^{-1}[(\mathscr{A} \otimes \mathbb{1}_2\mathbb{1}_2^\mathsf{T}) \odot \mathscr{M}]\mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}}$. Of course, by symmetry, $\epsilon_{i2} \equiv P(g_i^{\text{bin}}(\mathbf{x}) \leq \frac{m_{i1}+m_{i2}}{2} | \mathbf{x} \in \mathscr{C}_2)$ has the same limiting optimal value $\epsilon_{i2}^\star = \epsilon_{i1}^\star$.

The only non-diagonal matrices in (8) are Γ and $\mathscr{V}_i$ in which $\mathscr{M}$ plays the role of a "variance profile" matrix. In particular, assume $\Delta\mu_i^\mathsf{T}\Delta\mu_{i'} = 0$ for all $i' \neq i$, i.e., the differences in statistical means of all tasks are orthogonal to those of Task i. Then the two rows and columns of $\mathscr{M}$ associated to Task i are all zero but on the 2×2 diagonal block. Therefore, $y^{\text{bin}\star}$ will have all zero entries but on its Task i two elements. All other choices for the null entries of $y^{\text{bin}\star}$ (such as the usual $y^{\text{bin}} = [1, -1, \ldots, 1, -1]^\mathsf{T}$) 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 a targeted Task i, one must impose low absolute scores $y_{i'j}^{\text{bin}}$ to all Tasks i' strongly different from Task i.

The choice $y^{\text{bin}} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ can also be very detrimental when $\Delta\mu_i^\mathsf{T}\Delta\mu_{i'} < 0$ for some pair i, i': that is, when the mapping of the two classes within each task is reversed (e.g., if class $\mathscr{C}_1$ in Task 1 is closer to class $\mathscr{C}_2$ than class $\mathscr{C}_1$ in Task 2). In this setting, it is easily seen that $y^{\text{bin}} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ works against the classification and performs much worse than a single-task LSSVM.

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, $\boldsymbol{\delta}_{11}^{[k2]} = \ldots = \boldsymbol{\delta}_{k2}^{[k2]}$ and, since $y^{\text{bin}\star}$ is defined up to a multiplicative constant, we have

$$y^{\text{bin}\star} = \Gamma^{-1}\mathscr{V}_i^{-1}\left((\mathscr{A} \otimes \mathbb{1}_2\mathbb{1}_2^\mathsf{T}) \odot \mathscr{M}\right)(e_{i1}^{[2k]} - e_{i2}^{[2k]})$$

in which all matrices are organized in 2×2 blocks of equal entries. This immediately implies that $y_{i'1}^{\text{bin}\star} = -y_{i'2}^{\text{bin}\star}$ for all i'. So in particular, the detection threshold $\frac{1}{2}(m_{i1} + m_{i2})$ of the averaged-mean test (6) is zero (as conventionally assumed). In all other settings for the $n_{i'j}$'s, it is very unlikely that $y_{i1}^{\text{bin}\star} = -y_{i2}^{\text{bin}\star}$ and the optimal decision threshold *must* also be estimated. As a matter of fact, following up on Remark 3, the aforementioned optimal value $y^{\text{bin}\star}$ for y^{bin} is not unique and could be shifted by any constant vector. This extra degree of freedom will be of much relevance in the application Section 5, as commented in the following remark.

Remark 5 (Setting the decision threshold to zero) *As per Remark 3, the addition of a constant term to* y^{bin} *does not affect the ultimate performance of MTL-LSSVM. Yet, it affects the value of the limiting means* m_{ij} *of* $g_i^{\text{bin}}(\mathbf{x})$*, so in particular the value of the limiting optimal threshold* $\frac{1}{2}(m_{i1} + m_{i2})$*. Specifically, one may shift all entries of* y^{bin} *in such a way that* $\frac{1}{2}(m_{i1} + m_{i2}) = 0$ *and thus recenter the decision threshold to zero. For* $\bar{y} \in \mathbb{R}$ *this constant shift, this boils down to solving in the variable* $\bar{y}$ *the equation*

$$0 = \frac{1}{2}(m_{i1} + m_{i2}) = \frac{1}{2}(y^{\text{bin}} + \bar{y}e_i^{[k]} \otimes \mathbb{1}_2)^\mathsf{T}\mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{\frac{1}{2}}(I_{2k} - \mathscr{E}_e\Gamma)(e_{2(i-1)+1}^{[2k]} + e_{2i}^{[2k]})$$

where $\mathscr{E}_e = I_{2k} - \sum_{i'=1}^k E_{i'i'}^{[k]} \otimes c_{i'}$ *and* $c_{i'} = \mathbb{1}_2\left[\frac{n_{i'1}}{n_{i'}} \; \frac{n_{i'2}}{n_{i'}}\right]$*. Similarly, one may instead impose that* $m_{i1} = 0$*: this will appear to be fundamental to* align classifiers *in the multi-class "one-versus-all" extension of the present binary classification scheme (see details in Section 5.2).*

Remark 6 (Tuning the hyperparameters) *The previous section provided a high-level interpretation for the impact of the vector parameter $\gamma \in \mathbb{R}^k$ and the scalar parameter $\lambda \in \mathbb{R}$, the effect of which is to respectively regularize LSSVM learning and to set the throttle between individual versus collective learning. These hyperparameters intervene deeply inside our theoretical formulas (so far in Theorem 4 but later in Theorem 8) and are not amenable to simple optimization. Yet, as will be confirmed by experiments (see in particular Figure 3), the proposed optimization of the input scores $\mathscr{y}^{\mathrm{bin}}$ partly compensates for suboptimal choices in γ, λ. As such, an "informed guess", based on our previous discussion of the effects of these parameters, is in general sufficient for highly performing MTL-LSSVM. A further gradient descent operation (or local grid search) on the theoretical performance approximation, initialized at the informed guess values, can further improve the overall learning performance.*

3.3 Practical implementation of improved MTL-LSSVM

As already pointed out, a fundamental aspect of Theorem 4 lies in the performances of the *large dimensional* $(n, p \gg 1)$ classification problem at hand boiling down to $2k$-dimensional statistics. More importantly from a practical perspective, these $2k$-dimensional "sufficient statistics" are easily amenable to fast and efficient estimation: it indeed only requires a few training data samples to estimate all quantities involved in the theorem (which, as a corollary, lets one envision the possibility of efficient transfer learning methods based on very scarce data samples).

Remark 7 (On the estimation of m_{ij} and σ_i) *All quantities defined in Theorem 4 are a priori known, apart from the quantities $\mathscr{M} \equiv \sum_{i,i'} \Delta\mu_i^\mathsf{T} \Delta\mu_{i'} \left(E_{ii'}^{[k]} \otimes \mathbb{c}_i \mathbb{c}_{i'}^\mathsf{T} \right)$ and most specifically the inner products $\Delta\mu_i^\mathsf{T} \Delta\mu_{i'}$. For these, define, for $j = 1, 2$, two sets $\mathcal{S}_{ij}, \mathcal{S}'_{ij} \subset \{1, \ldots, n_{ij}\}$ and the corresponding indicator vectors $\mathbb{j}_{ij}, \mathbb{j}'_{ij} \in \mathbb{R}^{n_i}$ with $[\mathbb{j}_{ij}]_a = \delta_{a \in \mathcal{S}_{ij}}$ and $[\mathbb{j}'_{ij}]_a = \delta_{a \in \mathcal{S}'_{ij}}$. We further impose that $\mathcal{S}'_{ij} \cap \mathcal{S}_{ij} = \emptyset$. Then, for $i \neq i'$, the following estimates hold:*

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

Observe in particular that a single sample (two when $i = i'$) 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. In a transfer learning setting where some tasks may contain few labeled data, it is thus still possible to optimize the MTL algorithm. Of course, when more data are available, under our assumption that $p \sim n$, taking all samples in the averaging, the convergence speed is of order

$O(1/\sqrt{np}) = O(1/n)$, *which is a quadratic increase in the speed of the usual central-limit theorem.*

Estimating m_{ij} and σ_i not only allows one to anticipate theoretical performances but also enables the actual estimation of the decision threshold $\frac{1}{2}(m_{i1} + m_{i2})$ of the test (6) and, as shown previously, opens the possibility to largely optimize MTL-LSSVM through an (asymptotically) optimal choice of the training scores y^{bin}.

The series of theoretical and practical results of this section may be synthetized under the form of Algorithm 1.

Algorithm 1 Proposed binary Multi Task Learning algorithm.

Input: Training samples $X = [X_1, \ldots, X_k]$ with $X_{i'} = [X_{i'}^{(1)}, X_{i'}^{(2)}]$ and test data $\mathbf{x}$.
Output: Estimated class $\hat{j} \in \{1, 2\}$ of $\mathbf{x}$ for target Task i.
Center and normalize data per task: for all $i' \in \{1, \ldots, k\}$,

- $\mathring{X}_{i'} \leftarrow X_{i'} \left(I_{n_{i'}} - \frac{1}{n_{i'}} \mathbb{1}_{n_{i'}} \mathbb{1}_{n_{i'}}^{\mathsf{T}} \right)$
- $\mathring{X}_{i'} \leftarrow \mathring{X}_{i'} / \frac{1}{n_{i'} p} \text{tr}(\mathring{X}_{i'} \mathring{X}_{i'}^{\mathsf{T}})$

Estimate: Matrix M from Remark 7 and $\boldsymbol{\delta}^{[k]}$ by solving (3).
Create scores $y^{\text{bin}} = y^{\text{bin}\star}$ according to (8).
Compute the threshold ζ_i from (6), with m_{ij} defined in Theorem 4 for $y^{\text{bin}} = y^{\text{bin}\star}$.
(Optional) Estimate the theoretical classification error $\epsilon_{i1} = \epsilon_{i1}(\lambda, \gamma)$ from (7) and minimize over (λ, γ).[2]
Compute classification score $g_i(\mathbf{x})$ according to (2).
Output: $\hat{j}$ such that $g_i(\mathbf{x}) \underset{\hat{j}=2}{\overset{\hat{j}=1}{\gtrless}} \zeta_i$.

3.4 Empirical evidence

This section shortly illustrates the ideas and intuitions developed so far (such as the relevance of an optimal choice of the data labels and decision threshold) through the performances of Algorithm 1 on a transfer learning benchmark application. Sections 5–6 will cover a much larger spectrum of applications and experiments, under the most general data setting discussed in the subsequent sections.

For optimal comparison, we consider here the standard Office+Caltech256 real image classification benchmark (Saenko et al., 2010; Griffin et al., 2007), consisting of four tasks and $m = 10$ categories shared by all tasks. The dataset X consists here of the VGG features of size $p = 4096$ extracted from these images. We place ourselves under a $k = 2$ transfer learning setting where Task 1 is the source task and Task 2 is the target task (the performance of which we aim to optimize), taken from two of the four tasks of the

2. As per Remark 6, this operation involves reevaluating $\boldsymbol{\delta}^{[k]}$ and thus $y^\star$, and thus m for each (λ, γ). It can be performed either on a static grid or by gradient descent until a local minimum is reached.

dataset (Caltech, Webcam, Amason, dslr). For testing, the samples of the target task are randomly selected from the test dataset of Office+Caltech256 and the classification accuracy is averaged over 20 trials. Table 1 reports the accuracy for all possible pairs ($4 \times 3 = 12$ of them) of source and transfer tasks, obtained by Algorithm 1 (Ours) versus the non-optimized LSSVM of (Xu et al., 2013) (LSSVM) and versus other state-of-the-art transfer learning algorithms: the max margin domain transform of (Hoffman et al., 2013) (MMDT) which seeks a linear transform to match the source data to the target data and then applies an SVM on the resulting target domain; the cross-domain landmark selection (CDLS) of (Hubert Tsai et al., 2016), which learns a feature subspace which matches the cross-domain data distribution and eliminates the domain differences; and the invariant latent space (ILS) of (Herath et al., 2017), which, similar to MMDT, learns an invariant latent space in which the discrepancy between source and target is minimized. As already pointed out in introduction, since the article aims to propose an improved classification algorithm independent of the feature representation, it is fair to compare it to methods which use the same data features. The algorithms compared in the table all systematically use VGG features. It would be unfair to compare these against "end to end" MTL learning methods including a (explicit or implicit) step of feature learning like recent deep neural networks methods(Zhuang et al., 2020; Krishna and Kalluri, 2019).

Since $m = 10$ here, Algorithm 1 cannot rigorously be used as it stands. We apply instead a *naive* "one-versus-all" extension consisting in running in parallel $m = 10$ times Algorithm 1 by considering, for each class $\mathscr{C}_j$ of Task i, $1 \leq j \leq m$, a binary setting where the fictitious "Class $\tilde{\mathscr{C}}_1$" coincides with $\mathscr{C}_j$ and the second fictitious "Class $\tilde{\mathscr{C}}_2$" is the union of all $\mathscr{C}_{j'}$ for $j' \neq j$. Following up on Remark 5, each classifier $\ell \in \{1, \ldots, m\}$ can be set in such a way that $\mathbb{E}[g_i^{\text{bin}}(\mathbf{x}; \ell)] = 0$ when $\mathbf{x} \in \mathscr{C}_\ell$. For a new datum $\mathbf{x}$, of all m classifiers $g_i^{\text{bin}}(\mathbf{x}; 1), \ldots, g_i^{\text{bin}}(\mathbf{x}; m)$, the one reaching the greatest score is the selected allocation class for $\mathbf{x}$.

Table 1 demonstrates that our proposed improved MTL-LSSVM, despite its simplicity and unlike the competing methods used for comparison, has stable performances and is extremely competitive. It either outperforms all other methods or is second-to-best. But, most importantly, the method comes along with performance predictions and guarantees, which none of the competing works are able to provide.[3]

These preliminary results are already very conclusive and reveal the strength of our proposed methodology. Yet, the assumptions in place so far are restricted to random concentrated data with identity covariance and to a binary classification setting (which, as already observed, needs be adapted to account for more than two classes per task). The next sections elaborate on the more generic setting of $m \geq 2$ classes per task with more realistic data models. The theoretical results no longer reduce to compact expressions as in the previous sections but are easily understood having already delineated the main take-away messages and ideas.

3. In the present context of the *naive* "one-versus-all", this claim should be taken with care: the performance can indeed be predicted provided the binary class model $\mathcal{N}(\mu_{i1}, I_p)$ versus $\mathcal{N}(\mu_{i2}, I_p)$ correctly matches the actual data distribution; this is likely not the case here as the collected fictitious "$\tilde{\mathscr{C}}_2$" is rather a mixture of Gaussian rather than a unique Gaussian. In Section 5.2, a more elaborate, and theoretically better supported, version of the one-versus-all approach will be discussed.

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	w→ c	c→ a	a→ c	w→ a	a→ d	d→ a	w→ d	c→ d	d→ c	a→ w	d→ w	Mean 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**

4. The General Framework

The results from the previous section are extended here to the more realistic setting where the data arise from a mixture of $m \geq 2$ concentrated random vectors with generic covariance Σ_{ij}. New insights, and most importantly, more general and application-driven algorithms will be introduced. In addition, the results are presented here with a sketched development of their main technical arguments, the full proofs being deferred to the appendix.

4.1 Main ideas

Taking for the moment for granted the Gaussian limit for $g_i(\mathbf{x}) \in \mathbb{R}^m$ as $p, n \to \infty$ (for $1 \leq i \leq k$), the main technical task to obtain our main result (Theorem 8, which generalizes the already introduced Theorem 4) is to evaluate the large dimensional behavior $\boldsymbol{m}_{ij}$ and C_{ij} of the statistical mean $\mathbb{E}[g_i(\mathbf{x})] = \boldsymbol{m}_{ij} + o(1)$ and covariance matrix $\mathrm{Cov}[g_i(\mathbf{x})] = C_{ij} + o(1)$ of the classification score $g_i(\mathbf{x})$ in (2) for data vectors $\mathbf{x}$ in class $\mathcal{C}_j$ (i.e., such that $\mathbb{E}[\mathbf{x}] = \mu_{ij}$ and $\mathrm{Cov}[\mathbf{x}] = \Sigma_{ij}$), respectively given by:

$$\boldsymbol{m}_{ij} = \mathbb{E}\left[\frac{1}{kp}(Y - Pb)^\mathsf{T} Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} \left(e_i^{[k]} \otimes \mu_{ij}\right) + b_i\right] + o(1) \tag{10}$$

$$C_{ij} = \mathbb{E}\left[\frac{1}{(kp)^2}(Y - Pb)^\mathsf{T} Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z (Y - Pb)\right] + o(1) \tag{11}$$

with $S_{ij} = e_i^{[k]} e_i^{[k]\mathsf{T}} \otimes \Sigma_{ij}$ and $\tilde{Q} = \left(\frac{A^{\frac{1}{2}} Z Z^\mathsf{T} A^{\frac{1}{2}}}{kp} + I_{kp}\right)^{-1}$.

Our technical approach to evaluate these terms, in the large dimensional regime of Assumption 2 and for data distributed as per Assumption 1, consists in determining *deterministic equivalents*, a classical object in random matrix theory (Couillet and Debbah, 2011, Chapter 6), for the matrices $\tilde{Q}$, $\tilde{Q} A^{\frac{1}{2}} Z$, $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z$ which are at the core of the formulation of $\boldsymbol{m}_{ij}$ and C_{ij}. Specifically, a deterministic equivalent, say $\bar{F} \in \mathbb{R}^{n \times p}$, of a given random matrix $F \in \mathbb{R}^{n \times p}$ is a *deterministic matrix* such that, for any deterministic linear functional $f : \mathbb{R}^{n \times p} \to \mathbb{R}$ of bounded norm, $f(F - \bar{F}) \to 0$ almost surely – in particular, for u, v of unit norm, $u^\mathsf{T}(F - \bar{F})v \xrightarrow{\text{a.s.}} 0$ and, for $A \in \mathbb{R}^{p \times n}$ deterministic of bounded operator norm, $\frac{1}{n}\mathrm{tr} A(F - \bar{F}) \xrightarrow{\text{a.s.}} 0$. We will denote for short $F \leftrightarrow \bar{F}$ to indicate

that $\bar{F}$ is a deterministic equivalent for F. Deterministic equivalents are thus particularly suitable to handle bilinear forms involving the random matrix F, so in particular the statistics (10) and (11) of $g_i(\mathbf{x})$, seen as bilinear forms involving the random matrices $\tilde{Q}A^{\frac{1}{2}}Z$ and $Z^{\mathsf{T}}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}Z$.

Lemma 14, deferred to Section A.2 of the appendix (as the result in itself does not bring any deep insight worth discussing here), provides the necessary deterministic equivalents for these matrices. It is interesting to point out though that, from a technical standpoint, the block structure followed by the core data matrix Z introduced in Proposition 2 makes the large dimensional random matrix analysis more challenging and the result less straightforward than in similar previous works (Mai and Liao, 2019; Liao and Couillet, 2019). Even in the simplest setting where the $x_{il}^{(j)}$ would be vectors of i.i.d. $\mathcal{N}(0,1)$ entries, the matrix Z is *not* a matrix of i.i.d. entries (due to precisely located blocks of zeros) and the singular values of Z do not asymptotically follow the popular Marčenko-Pastur distribution from (Marčenko and Pastur, 1967), as would be the case in works dealing with single-task learning (single-task LSSVM (Liao and Couillet, 2019), semi-supervised learning (Mai and Liao, 2019), neural networks (Louart et al., 2018), etc.).

The main information to be extracted from Lemma 14 (again, see its complete form in the appendix) is the central role played by the deterministic matrices

$$M = \left(e_1^{[k]} \otimes [\mu_{11}, \ldots, \mu_{1m}], \ldots, e_k^{[k]} \otimes [\mu_{k1}, \ldots, \mu_{km}]\right)$$
$$\mathbb{C}_{ij} = A^{\frac{1}{2}} \left(e_i^{[k]} {e_i^{[k]}}^{\mathsf{T}} \otimes (\Sigma_{ij} + \mu_{ij}\mu_{ij}^{\mathsf{T}}) \right) A^{\frac{1}{2}}$$

which generalize the matrices $\mathscr{M}$ and $\mathscr{A}$ discussed at length in Section 3 when $\Sigma_{ij} = I_p$. While gaining in genericity, unlike $\mathscr{M}$, the matrices M and $\mathbb{C}_{ij}$ preserve their large dimensions: this is the main price paid by the generalization to $\Sigma_{ij} \neq I_p$. Yet, the central small dimensional matrix Γ defined in (5) remains small and now becomes

$$\Gamma = \left(I_{mk} + \mathbb{M}^{\mathsf{T}} \bar{\tilde{Q}}_0 \mathbb{M}\right)^{-1}$$
$$\bar{\tilde{Q}}_0 = \left[\sum_{i=1}^{k} \sum_{j=1}^{m} (\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k)^{\frac{1}{2}} e_i^{[k]} {e_i^{[k]}}^{\mathsf{T}} (\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k)^{\frac{1}{2}} \otimes \boldsymbol{\delta}_{ij}^{[mk]} \Sigma_{ij} + I_{kp} \right]^{-1}$$
$$\mathbb{M} = A^{\frac{1}{2}} M \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}$$

and the mk scalars $\boldsymbol{\delta}_{ij}^{[mk]}$ are the unique positive solutions of the fixed point equations

$$\boldsymbol{\delta}_{ij}^{[mk]} = \frac{c_{ij}}{c_0 \left(1 + \frac{1}{kp} \mathrm{tr}(\mathbb{C}_{ij} \bar{\tilde{Q}})\right)}$$
$$\bar{\tilde{Q}} = \left(\sum_{i=1}^{k} \sum_{j=1}^{m} \boldsymbol{\delta}_{ij}^{[mk]} \mathbb{C}_{ij} + I_{kp} \right)^{-1}.$$

Here, $\bar{\tilde{Q}}$ is a deterministic equivalent of $\tilde{Q}$. Finally, the matrix $\mathscr{K}$ appearing in the variance term of Theorem 4 now becomes

$$\begin{aligned}\mathscr{K} &= c_0\bar{T}\left(\mathscr{D}_c - c_0\mathscr{T}\right)^{-1}\\ \bar{T}_{ij,i'j'} &= \frac{\boldsymbol{\delta}_{ij}^{[mk]}\boldsymbol{\delta}_{i'j'}^{[mk]}}{kp}\mathrm{tr}\left(\mathbb{C}_{i'j'}\bar{\tilde{Q}}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\bar{\tilde{Q}}\right)\\ \mathscr{T}_{ij,i'j'} &= \frac{\boldsymbol{\delta}_{ij}^{[mk]}\boldsymbol{\delta}_{i'j'}^{[mk]}}{kp}\mathrm{tr}(\mathbb{C}_{ij}\bar{\tilde{Q}}\mathbb{C}_{i'j'}\bar{\tilde{Q}})\end{aligned}$$

where $\bar{T}_{ij,i'j'}$ is the element at row $m(i-1)+j$ and column $m(i'-1)+j'$ of $\bar{T}$ (and similarly for $\mathscr{T}$) and $\kappa_{ij,\cdot} \in \mathbb{R}^{mk}$ represents the $m(i-1)+j$ row of matrix $\kappa \in \mathbb{R}^{mk\times mk}$.

With these technical elements at hand, we are in position to enunciate the main result of the article.

4.2 Classification score asymptotics

Theorem 8 *Under Assumptions 1 and 2, for a test data* $\mathbf{x}$ *with* $\mathbb{E}[\mathbf{x}] = \mu_{ij}$ *and* $\mathrm{Cov}[\mathbf{x}] = \Sigma_{ij}$*, as* $p, n \to \infty$*,*

$$g_i(\mathbf{x}) - G_{ij} \to 0, \quad G_{ij} \sim \mathcal{N}(\boldsymbol{m}_{ij}, C_{ij})$$

in law where, letting $\boldsymbol{m} = [\boldsymbol{m}_{11}, \ldots, \boldsymbol{m}_{km}]^\mathsf{T} \in \mathbb{R}^{km\times m}$ *and the normalized forms* $\boldsymbol{\mathcal{Y}} \equiv \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\mathcal{Y}$, $\mathring{\boldsymbol{\mathcal{Y}}} = \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\mathring{\mathcal{Y}}$, $\boldsymbol{m} = \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\boldsymbol{m}$,

$$\begin{aligned}\boldsymbol{m} &= \boldsymbol{\mathcal{Y}} - \Gamma\mathring{\boldsymbol{\mathcal{Y}}} \in \mathbb{R}^m\\ C_{ij} &= \mathring{\boldsymbol{\mathcal{Y}}}^\mathsf{T}\Gamma\mathscr{V}_{ij}\Gamma\mathring{\boldsymbol{\mathcal{Y}}} \in \mathbb{R}^{m\times m}\end{aligned}$$

with

$$\begin{aligned}\mathscr{V}_{ij} &= \mathscr{D}_{\kappa_{ij,\cdot}} + \mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0\mathbb{V}_{ij}\bar{\tilde{Q}}_0\mathbb{M}\\ \mathbb{V}_{ij} &= A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}} + \sum_{i'=1}^{k}\sum_{j'=1}^{m}\boldsymbol{\delta}_{i'j'}^{[mk]}\kappa_{ij,i'j'}A^{\frac{1}{2}}S_{i'j'}A^{\frac{1}{2}}\\ \kappa_{ij,i'j'} &= \frac{\mathscr{K}_{ij,i'j'}}{\boldsymbol{\delta}_{ij}^{[mk]}}.\end{aligned}$$

Proof See Section A.3 of the appendix. ■

In the particular case of $\Sigma_{ij} = I_p$ and $m = 2$, Theorem 8 reduces to Theorem 4 (see details in Section A.3 of the appendix) by remarking that $\mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0\mathbb{M} = \left(\mathscr{A}\otimes\mathbb{1}_k\mathbb{1}_k^\mathsf{T}\right)\odot\mathscr{M}$ and $\mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0V_{ij}\bar{\tilde{Q}}_0\mathbb{M} = \left(\mathscr{A}\mathscr{D}_{\kappa_i+e_i^{[k]}}\mathscr{A}\otimes\mathbb{1}_k\mathbb{1}_k^\mathsf{T}\right)\odot\mathscr{M}$ which, as already pointed out, have the advantage to be defined as the product of exclusively small dimensional matrices. Still, although more technical, Theorem 8 follows the same structure as Theorem 4.

Before concretely applying the result of Theorem 8 to practical learning problems (multi-task, transfer learning, hypothesis testing), a few comments and immediate corollaries are in order.

Remark 9 (Optimization of $\mathscr{y}^{\text{bin}}$ for $m = 2$ and generic Σ_{ij}) *As suggested in Section 3, for binary classification ($m = 2$), it is particularly convenient to recast the score vectors $y_{il}^{(j)} \in \mathbb{R}^m$ into scalar scores $y_{il}^{\text{bin}(j)} \in \mathbb{R}$ (this being irrespective of the nature of Σ_{ij}). Inspired by Section 3, one can trivially extend Theorem 8 to this binary setting. In this case, $g_i(\mathbf{x}) \in \mathbb{R}^m$ is now turned into a scalar $g_i^{\text{bin}}(\mathbf{x}) \in \mathbb{R}$ well approximated by $\mathcal{N}(m_{ij}, C_{ij})$ where now m_{ij} and C_{ij} are scalar, obtained by simply replacing $y_{il}^{(j)} \in \mathbb{R}^m$ by $y_{il}^{\text{bin}(j)} \in \mathbb{R}$ in their respective expressions.*

With these notations, setting the decision threshold of $g_i^{\text{bin}}(\mathbf{x})$ to $\zeta \in \mathbb{R}$ and assuming equal prior probability for the genuine class of $\mathbf{x}$, the classification error rate for a target task i is

$$E = \frac{1}{2}\mathcal{Q}\left(\frac{\zeta - m_{i1}}{\sqrt{C_{i1}}}\right) + \frac{1}{2}\mathcal{Q}\left(\frac{\zeta - m_{i2}}{\sqrt{C_{i2}}}\right).$$

As in Section 3, if $\Sigma_{i1} = \Sigma_{i2}$, then $C_{i1} = C_{i2} \equiv C_i$ and the decision threshold ζ minimizing the classification error is

$$\zeta^\star = \frac{m_{i1} + m_{i2}}{2}$$

from which the optimal vector $\mathscr{y}^{\text{bin}}$ for Task i is computed as

$$\begin{aligned}\mathscr{y}^{\text{bin}\star} &= \underset{\mathscr{y}^{\text{bin}} \in \mathbb{R}^{2k}}{\arg\max} \frac{(m_{i1} - m_{i2})^2}{C_i} \\ &= \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{-\frac{1}{2}} \Gamma^{-1} \mathscr{V}_{ij}^{-1} \mathbb{M}^\mathsf{T} \tilde{\bar{Q}}_0 \mathbb{M} \mathscr{D}_{\boldsymbol{\delta}^{[2k]}}^{-\frac{1}{2}} (e_{i1}^{[2k]} - e_{i2}^{[2k]}). \end{aligned} \tag{12}$$

It is important to recall here that, while $\mathscr{y}^{\text{bin}\star}$ expresses here solely as a function of terms involving the index i, all other statistics of the tasks $i' \neq i$ are in fact "embedded" inside these terms and are thus, of course, accounted for in the optimization.

When $C_{i1} \neq C_{i2}$ (which is the case in general), one may minimize E by resorting to numerical optimization techniques. We suggest to use a gradient descent method initialized to the expression obtained in (12). So long that Σ_{i1} and Σ_{i2} are not drastically different, this approach shows good performances (see our results in Section 6).

This said, the specific setting of binary classification may in practice be one of hypotheses testing. Under this scenario, one may not demand that the average error E be minimized (i.e., that data from either class is equally well identified) but rather that the probability of misclassification of a given class (say, a Type-I error) be bounded to some $\eta > 0$ while minimizing the error rate for the other class (Type-II error). In this context, if, say, one fixes $\mathcal{Q}(\frac{\zeta - m_{i1}}{\sqrt{C_{i1}}}) \equiv \eta$, then the classification error for the second class $\mathcal{Q}(\frac{\zeta - m_{i2}}{\sqrt{C_{i2}}})$ is minimized by now choosing

$$\mathscr{y}^{\text{bin}\star} = \underset{\mathscr{y}^{\text{bin}} \in \mathbb{R}^{2k}}{\arg\max} \frac{(\sqrt{C_{i1}}\mathcal{Q}^{-1}(\eta) + m_{i1} - m_{i2})^2}{C_{i2}}$$

where $\mathcal{Q}^{-1}$ is the inverse function of the $\mathcal{Q}$ function. This again can be solved using numerical convex optimization initialized at the value of (12). More details on this hypotheses testing setting, along with concrete experiments, are developed in Section 6.

Remark 10 (Estimation of m_{ij} and C_{ij}) *In order to anticipate the performances and best set the decision thresholds for classification, one needs to access all quantities arising in*

Theorem 8. Yet, as opposed to Remark 7, where the low dimensional quantities of interest (mainly the inner products between statistical means) are easily estimated, the low dimensional quantities involved in Theorem 8 are less convenient to estimate, this being due to the presence of the a priori unknown covariance matrices Σ_{ij}. We propose here two strategies:

1. *either make the assumption that $\Sigma_{ij} \simeq \alpha_{ij} I_p$ with α_{ij} estimated by $\frac{1}{pn_{ij}} \mathrm{tr} \mathring{X}_i^{(j)} \mathring{X}_i^{(j)\mathsf{T}}$; then normalize the data as $\mathring{X}_i^{(j)} \leftarrow \mathring{X}_i^{(j)} / \frac{1}{pn_{ij}} \mathrm{tr} \mathring{X}_i^{(j)} \mathring{X}_i^{(j)\mathsf{T}}$ in the spirit of Algorithm 1. This places the experimenter under an isotropic data setting for all classes and tasks, from which the considerations of Section 3 (possibly generalized to $m > 2$) apply.*

2. *either estimate Σ_{ij} by means of the sample covariance matrix $\frac{1}{n_{ij}} \mathring{X}_i^{(j)} \mathring{X}_i^{(j)\mathsf{T}}$; this procedure is known to be biased, particularly so if n_{ij} is not large compared to p; yet, as demonstrated in our experiments in Section 6, this only marginally (if not at all) alters the performance of our proposed algorithms.*[4]

The choice of strategy mainly depends on the belief from the experimenter that the genuine covariance matrices are "well-conditioned" (i.e., their eigenvalues do not spread much) in which case Option 1 would be favored or "ill-conditioned" (typically when the space spanned by the data is much lower than p) in which case Option 2 would be more appropriate.

Remark 11 (On universality) *As pointed out in the introduction, the input data X follows a very generic* concentrated random vector *assumption (Assumption 1). This choice provides both a technical, but most importantly, a fundamentally practical, advantage:*

1. *from a technical standpoint, the concentration of measure phenomenon provides efficient and fast mathematical tools (Louart and Couillet, 2018; Ledoux, 2001) to analyze the random quantities appearing in the classification test scores $g_i(\mathbf{x})$ of MTL-LSSVM (which, in essence, is a mere functional $\mathbb{R}^{p \times n} \to \mathbb{R}$ of the random input data X). More specifically, alternative random matrix tools based on Gaussian (Pastur and Shcherbina, 2011) or independent entries assumptions (Bai and Silverstein, 2009) of X would both be less general (at least for our machine learning purpose) and more computationally intense;*

2. *on the practical side, as underlined in Section 2.2, the concentrated random vector assumption better models realistic datasets by imposing very little structure on the data. Exactly, it only constrains all* Lipschitz functionals $\mathbb{R}^{p \times n} \to \mathbb{R}$ *of X (i.e., its typical observations) to satisfy a concentration inequality; while this may seem demanding, the family of concentrated random vectors in fact contains all Lipschitz generative models (for instance neural networks) fed by Gaussian inputs (such as GANs (Goodfellow et al., 2014)), as well as all further Lipschitz transformations of these vectors (for instance, features extracted by pretrained neural networks). As such, provided that the assumption of a common statistical mean and covariance per class*

4. It must be pointed out that similar random matrix-based studies propose consistent estimates for low dimensional quantities such as those met in Theorem 8; however, these would assume cumbersome forms which, we believe, go against our present request for simple, intuitive but well parameterized algorithms for multi-task and transfer learning.

and per task is reasonable, Theorem 8 ensures for instance that the performance of MTL-LSSVM applied to classes of the popular VGG or ResNet representations of GAN images is predictable. From this remark, it naturally comes that the proposed method is universal *in the sense of its being robust to a broad range of very realistic random data, and it is not daring to claim that it is equally valid on genuinely real data. This is confirmed by our numerical results of Section 6.*

With these elements in place, we are in position to apply our findings to a host of applications in statistical learning and to test the resulting algorithms against state-of-the-art alternatives.

5. Applications

This section provides various applications and optimizations of the proposed MTL-LSSVM framework based on the findings of the previous sections in the context of multi-class classification.

Having access to the large dimensional behavior of the classification test score in Theorem 8 (i.e., for $m \geq 2$ classes per task) is more fundamental than one may think. It indeed allows for a fine-tuning of the hyperparameters to be set to extend the usually considered binary MTL framework of (Evgeniou and Pontil, 2004; Xu et al., 2013) to a multiclass-per-task MTL.

5.1 Multi-class classification preliminary

The literature (Bishop, 2006; Rocha and Goldenstein, 2013) describes broad groups of approaches for dealing with $m > 2$ classes. We focus here on the most common methods, namely one-versus-all, one-versus-one, and one hot encoding. Being so far theoretically intractable (before the results of this article), these methods inherently suffer from sometimes severe limitations; these are partly tackled by adapting the theoretical results discussed in Section 4:

1. **one-versus-all**: in this method, focusing on Task i, m individual binary classifiers $g_i^{\text{bin}}(\ell)$ for $\ell = 1, \ldots, m$ are trained, each of them separating Class $\mathscr{C}_\ell$ from the other $m-1$ classes $\mathscr{C}_{\ell'}$, $\ell' \neq \ell$. Each test sample is then allocated to the class with the highest score among the m classifiers. Although quite used in practice, the approach first suffers a severe data unbalancing effect when using binary (± 1) labels as the set of negative labels in each binary classification is on average $m-1$ times larger than the set of positive labels, and also suffers a centering-scale issue when ultimately comparing the outputs of the m decision functions $g_i^{\text{bin}}(\mathbf{x}; \ell)$, $\ell = 1, \ldots, m$, whose average locations and ranges may greatly differ; these issues lead to undesirable effects, as reported in (Bishop, 2006, section 7.1.3)).

   In Section 5.2, these problems are simultaneously addressed: specifically, having access to the theoretical statistics of the classification scores allows us to appropriately center and scale the scores. Moreover, each binary classifier is optimized by appropriately choosing the class labels (no longer binary) so to minimize the resulting classification error (see Figure 1 for an illustration of the improvement induced by the proposed approach).

2. **one-versus-one**: here, $\frac{1}{2}m(m-1)$ binary classifiers are trained (one for each pair j, j' of classes, solving a binary classification). For each test sample, each binary classifier decides on – or "votes" for – the more relevant class. The test sample is then attributed to the class having the majority of votes. Although the number of binary classifiers is greater than in the one-versus-all approach, the training process for each classifier is faster since the training database is much smaller for each binary classifier. Besides, the method is more robust to class imbalances (since only pairwise comparisons are made) but suffers from an undecidability limitation in the case of equal numbers of majority votes for two or more classes.

   In Section 5.3, each binary classifier will be optimized according to Algorithm 1 by choosing appropriate labels and appropriate decision thresholds, thereby largely improving over the basal classifier performance.

3. **one-hot encoding approach**, also known as one-per-class coding: in this approach, each class is encoded using the m-dimensional canonical vector of the class (the code vector for class j has a 1 at position j and 0's elsewhere). When testing an unknown sample $\mathbf{x}$, the index of the encoding output vector $g_i(\mathbf{x}) \in \mathbb{R}^m$ with maximum value is selected as the class of $\mathbf{x}$.

   Exploiting the asymptotic performance of this approach from Theorem 8 allows us to derive a different label (or score) encoding for each class which theoretically minimizes the classification error. This is developed in detail in Section 5.4.

In the remainder of the section, each of the three classifiers is studied, optimized and their asymptotic performances are analyzed according to our previous results except for one-versus-one classification which involves difficult combinatorial aspects. While this does not provide a definite and general answer as to which of the three classifiers is best, it however provides an accurate assessment of their asymptotic performances; most importantly, these performances may be evaluated *before* running the classifier, thereby helping practitioners to anticipate and optimize the method best suited for the application at hand, without resorting to any cross-validation procedure.

Let us finally insist that, for the two multi-class extensions based on binary classifiers (one-versus-one, one-versus-all), each binary classifier will be optimized independently following Remark 9: i.e., by recasting the score vectors $y_{il}^{(j)} \in \mathbb{R}^m$ into scalar scores $y_{il}^{\mathrm{bin}(j)} \in \mathbb{R}$. As such, from now on, for each binary classifier ℓ, $g_i(\mathbf{x}; \ell) \in \mathbb{R}^m$ will be systematically turned into a scalar $g_i^{\mathrm{bin}}(\mathbf{x}, \ell) \in \mathbb{R}$ well approximated by $\mathcal{N}(m_{ij}, C_{ij})$ where now m_{ij} and C_{ij} are scalar, obtained by simply replacing $y_{il}^{(j)}(\ell) \in \mathbb{R}^m$ by $y_{il}^{\mathrm{bin}(j)}(\ell) \in \mathbb{R}$ in their respective expressions.

5.2 One-versus-all multi-class classification

For every Task i, the one-versus-all approach solves m binary MTL-LSSVM algorithms with target class $\mathscr{C}_\ell$, for each $\ell \in \{1, \ldots, m\}$, versus all other classes $\mathscr{C}_{\ell'}$, $\ell' \neq \ell$. Calling $g_i^{\mathrm{bin}}(\mathbf{x}; \ell)$ the output of the classifier ℓ for a new datum $\mathbf{x}$, the class allocation decision is traditionally based on the largest among all scores $g_i^{\mathrm{bin}}(\mathbf{x}; 1), \ldots, g_i^{\mathrm{bin}}(\mathbf{x}; m)$. This approach generalizes the naive, yet simpler, method proposed in Algorithm 1 which, despite its good

performances (recall Table 1), is fundamentally "incorrect" in its assuming that, for each ℓ, all classes $\mathscr{C}_{\ell'}$ ($\ell' \neq \ell$) have the same statistics.

However, this presumes that the distribution of the scores $g_i^{\text{bin}}(\mathbf{x};1)$ when $\mathbf{x} \in \mathscr{C}_1$, $g_i^{\text{bin}}(\mathbf{x};2)$ when $\mathbf{x} \in \mathscr{C}_2$, etc., have more or less the same mean and variance. This is not the case in general, as depicted in the first column of Figure 1, where data from class $\mathscr{C}_1$ are more likely to be allocated to class $\mathscr{C}_3$ (compare the red curves).

By providing an accurate estimate of the distribution of the scores $g_i^{\text{bin}}(\mathbf{x};\ell)$ for all ℓ and all genuine classes of $\mathbf{x}$, Theorem 8 allows us to predict the various positions of the Gaussian curves in Figure 1. In particular, exploiting the theorem along with Remark 5, it is possible, for binary classifier ℓ to shift the corresponding input scores $\mathcal{y}^{\text{bin}}(\ell)$ by a constant term $\bar{\mathcal{y}}(\ell) \in \mathbb{R}$ in such a way that $\mathbb{E}_{\mathbf{x}\in\mathscr{C}_\ell}[g_i^{\text{bin}}(\mathbf{x};\ell)] \simeq \mathcal{m}_{i\ell}(\ell) = 0$ and $\text{Var}_{\mathbf{x}\in\mathscr{C}_\ell}[g_i^{\text{bin}}(\mathbf{x};\ell)] \simeq C_{i\ell}(\ell) = 1$. This operation prevents the centering and scale problems depicted in the first column of Figure 1, the result being visible in the second column of Figure 1.

This first improvement step simplifies the algorithm which still boils down to determining the largest $g_i^{\text{bin}}(\mathbf{x};\ell)$, $\ell \in \{1,\dots,m\}$, output but now limiting the risks induced by the inherent centering and scale issues previously discussed.

This being said, our theoretical analysis further allows to adapt the input scores $\mathcal{y}^{\text{bin}}(\ell)$ in such a way to optimize the expected output. Ideally, assuming $\mathbf{x}$ genuinely belongs to class ℓ, one may aim at increasing the distance between the output score $g_i^{\text{bin}}(\mathbf{x};\ell)$ and the other output scores $g_i^{\text{bin}}(\mathbf{x};\ell')$ for $\ell' \neq \ell$. This however demands to simultaneously adapt all input scores $\mathcal{y}^{\text{bin}}(1),\dots,\mathcal{y}^{\text{bin}}(m)$. Instead, we resort to maximizing the distance between the output score $g_i^{\text{bin}}(\mathbf{x};\ell)$ for $\mathbf{x} \in \mathscr{C}_\ell$ and the scores $g_i^{\text{bin}}(\mathbf{x};\ell)$ for $\mathbf{x} \notin \mathscr{C}_\ell$. By "mechanically" pushing away all wrong decisions, this ensures that, when $\mathbf{x} \in \mathscr{C}_\ell$, $g_i^{\text{bin}}(\mathbf{x};\ell)$ is greater than $g_i^{\text{bin}}(\mathbf{x};\ell')$ for $\ell' \neq \ell$. This is visually seen in the third column of Figure 1, where the distances between the rightmost Gaussians and the other two is increased when compared to the second column, and we retrieve the desired behavior.

Specifically, our proposed score optimization consists in solving, for each $i \in \{1,\dots,k\}$ and each $\ell \in \{1,\dots,m\}$ the optimization problems:

$$\begin{aligned}
\mathcal{y}^{\text{bin}\star}(\ell) &= \underset{\mathcal{y}^{\text{bin}}(\ell)\in\mathbb{R}^{km}}{\arg\min} \max_{j\neq\ell} \mathcal{Q}\left(\frac{\mathcal{m}_{i\ell}(\ell) - \mathcal{m}_{ij}(\ell)}{\sqrt{C_{tj}}}\right) \\
&= \underset{\mathcal{y}^{\text{bin}}(\ell)\in\mathbb{R}^{km}}{\arg\min} \max_{j\neq\ell} \mathcal{Q}\left(\frac{\mathcal{y}^{\text{bin}}(\ell)^\mathsf{T}\left(I_{mk} - \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{-\frac{1}{2}}\Gamma\mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\right)(e_{m(i-1)+\ell}^{[mk]} - e_{m(i-1)+j}^{[mk]})}{\sqrt{\mathcal{y}^{\text{bin}}(\ell)^\mathsf{T}\mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\Gamma\mathscr{V}_{ij}\Gamma\mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}\mathcal{y}^{\text{bin}}(\ell)}}\right)
\end{aligned} \tag{13}$$

with $\mathcal{Q}$ the Gaussian q-function.

Being a non-convex and non-differentiable (due to the max) optimization, Equation 13 cannot be solved straightforwardly. An approximated solution consists in relaxing the max operator $\max(x_1,\dots,x_n)$ into the differentiable operator $\frac{1}{\gamma n}\log(\sum_{j=1}^n \exp(\gamma x_j))$ for some $\gamma > 0$, and use a standard gradient descent optimization scheme here initialized at $\mathcal{y}^{\text{bin}}(\ell) \in \mathbb{R}^{mk}$ filled with 1's at every $m(i'-1)+\ell$, for $i' \in \{1,\dots,m\}$, and with -1's everywhere else.

In effect, the optimized vector $\boldsymbol{y}^{\text{bin}\star}(\ell)$ is evaluated first *before* the constant shift scalar $\bar{\boldsymbol{y}}$ (ensuring that $\mathbb{E}_{\mathbf{x}\in\mathscr{C}_\ell}[g_i^{\text{bin}}(\mathbf{x};\ell)]$ is close to zero) is added to $\boldsymbol{y}^{\text{bin}\star}(\ell)$. This order of treatment is mandatory as $\mathbb{E}_{\mathbf{x}\in\mathscr{C}_\ell}[g_i^{\text{bin}}(\mathbf{x};\ell)]$ depends explicitly on the value of the input score vector $\boldsymbol{y}^{\text{bin}}$. This global procedure is described in Algorithm 2 below.

Algorithm 2 Proposed one-versus-all multi-task learning algorithm.

Input: Training samples $X = [X_1, \ldots, X_k]$ with $X_i = [X_i^{(1)}, \ldots, X_i^{(m)}]$, $X_i^{(j)} \in \mathbb{R}^{p\times n_{ij}}$ and test data $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for Task i.
for $\ell = 1$ **to** m **do**
Center and normalize data per task: for all $i' \in \{1, \ldots, k\}$,

- $\mathring{X}_{i'} \leftarrow X_{i'}\left(I_{n_{i'}} - \frac{1}{n_{i'}}\mathbb{1}_{n_{i'}}\mathbb{1}_{n_{i'}}^{\mathsf{T}}\right)$
- $\mathring{X}_{i'} \leftarrow \mathring{X}_{i'}/\frac{1}{n_{i'}p}\text{tr}(\mathring{X}_{i'}\mathring{X}_{i'}^{\mathsf{T}})$.

Estimate: $\mathbb{M}^{\mathsf{T}}\bar{\tilde{Q}}_0\mathbb{M}$ and $\mathscr{V}_{i\ell}$ according to Remark 10.
Create scores $\boldsymbol{y}^{\text{bin}\star}(\ell)$ by numerically solving (13) (see discussion following (13)).
Shift scores $\boldsymbol{y}^{\text{bin}\star}(\ell)$ according to Remark 5.
Estimate $C_{i\ell}(\ell)$ from Theorem 8 and Remark 10.
Compute classification scores $g_i^{\text{bin}}(\mathbf{x};\ell)$ according to (2).
end for
Output: $\hat{\ell} = \arg\max_{\ell\in\{1,\ldots,m\}}\left\{\frac{g_i^{\text{bin}}(\mathbf{x};\ell)}{\sqrt{C_{i\ell}(\ell)}}\right\}$.

As an immediate corollary of Theorem 8, for large dimensional data, the classification accuracy of Algorithm 2 can be precisely estimated, as follows.

Proposition 12 *Under the notations of Theorem 8, the probability of correct classification* $P_i^{(j)}(\mathbf{x})$ *for Task* i *of a test data* $\mathbf{x} \in \mathscr{C}_j$ *is given by*

$$P_i^{(j)}(\mathbf{x}) = \underbrace{\int\cdots\int_0^\infty}_{m-1} \frac{1}{\sqrt{(2\pi)^{m-1}|C(j)|}} \exp\left(-\frac{1}{2}(x-\mu(j))^{\mathsf{T}}C(j)^{-1}(x-\mu(j))\right)dx + o(1)$$

where $\mu(j) = \mathscr{Y}_{-j}(I_{mk} - \mathscr{D}_{\boldsymbol{\delta}}^{-\frac{1}{2}}\Gamma\mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}})e_{m(i-1)+j}^{[mk]} \in \mathbb{R}^{m-1}$ *and* $C(j) = \mathscr{Y}_{-j}\mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}}\Gamma\mathscr{V}_{ij}\Gamma\mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}}\mathscr{Y}_{-j}^{\mathsf{T}} \in \mathbb{R}^{(m-1)\times(m-1)}$, *with* $\mathscr{Y}_{-j} = \{\boldsymbol{y}^{\text{bin}}(j)^{\mathsf{T}} - \boldsymbol{y}^{\text{bin}}(j')^{\mathsf{T}}\}_{j'\neq j} \in \mathbb{R}^{(m-1)\times km}$.

Figure 1, succinctly introduced above, illustrates the successive improvements of the proposed algorithms. Specifically, it shows the gains of the centering-scaling operation on the input and output scores (2nd column) and of the optimization of the input scores (3rd column) when compared with the standard approach (1st column). Here synthetic data arising from a Gaussian mixture model are considered in a two-task ($k = 2$) and three-class ($m = 3$) setting in which $x_{1l}^{(j)} \sim \mathcal{N}(\mu_{1j}, I_p)$ and $x_{2l}^{(j)} \sim \mathcal{N}(\mu_{2j}, I_p)$, where $\mu_{2j} = \beta\mu_{1j} + \sqrt{1-\beta^2}\mu_{1j}^{\perp}$, with $\mu_{1j} = 2e_j^{[p]}$ and $\mu_{1j}^{\perp} = e_{p-j}^{[p]}$. Here $p = 200$, $\beta = 0.2$,

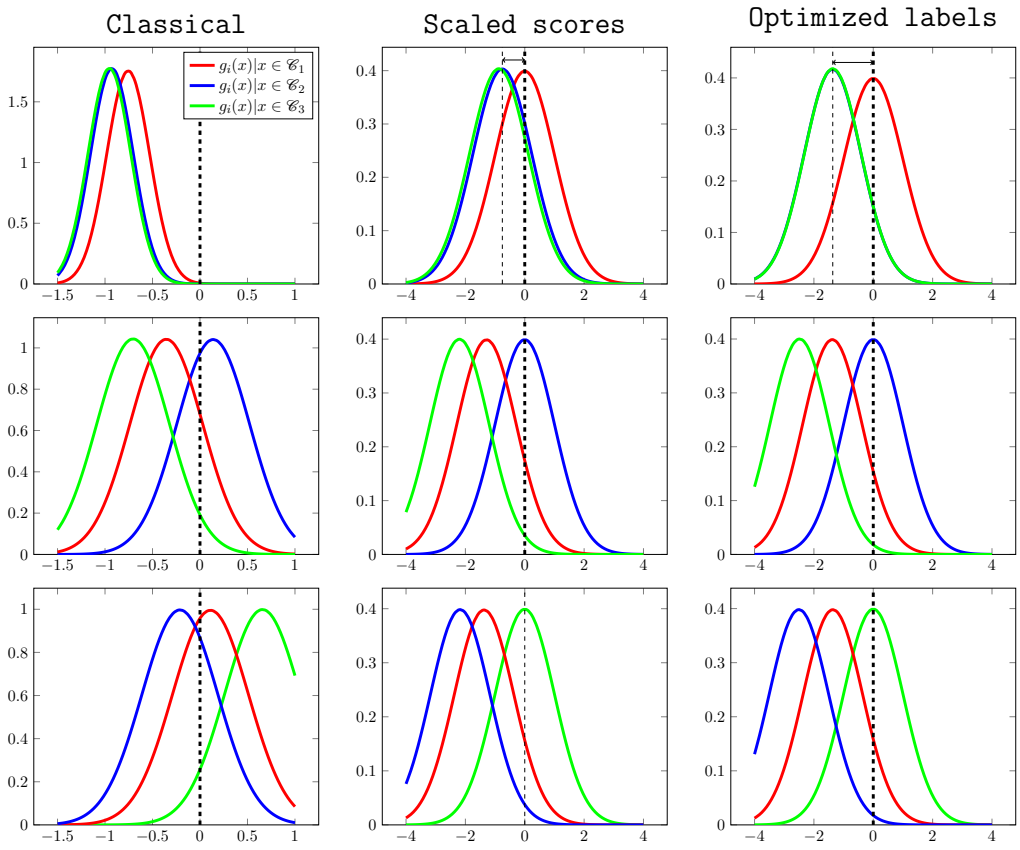

Figure 1: Test score distribution in a 2-task and 3 classes-per-task setting, using a one-versus-all multi-class classification. Every graph in row ℓ depicts the limiting distributions of $g_i(\mathbf{x};\ell)$ for $\mathbf{x}$ in different classes. Column 1 (Classical) is the standard implementation of the one-versus-all approach. Column 2 (Scaled scores) is the output for centered and scaled $g_i(\mathbf{x};\ell)$ for $\mathbf{x} \in \mathscr{C}_\ell$. Column 3 (Optimized labels) is the same as Column 2 but with optimized input scores (labels) $\mathscr{y}^{\text{bin}\star}(\ell)$. Under "classical" approach, data from $\mathscr{C}_1$ (red curves) will often be misclassified as $\mathscr{C}_2$. With "optimized labels", the discrimination of scores for $\mathbf{x}$ in either class $\mathscr{C}_2$ or $\mathscr{C}_3$ is improved (blue curve in 2nd row further away from blue curve in 1st row; and similarly for green curve in 3rd versus 1st row).

$[n_{11}, n_{12}, n_{13}, n_{21}, n_{22}, n_{23}] = [393, 309, 394, 20, 180, 480]$ and the optimization framework used for input score (label) $\mathscr{y}^{\text{bin}}$ optimization is a standard interior point method (Boyd and Vandenberghe, 2004).[5]

5. Here we use the `fmincon` function implemented in Matlab.

5.3 One-versus-one multi-class classification

For a given Task i, the one-versus-one multi-class method trains $\frac{1}{2}m(m-1)$ binary classifiers for each pair $j \neq j' \in \{1, \ldots, m\}$. As intensively discussed in the previous section, as well as in Section 3 and Remark 9, each resulting binary classifier $g_i^{\text{bin}}(\mathbf{x}; j, j')$ can be optimized by choosing optimal input labels $\mathscr{y}^{\text{bin}}(j, j')$. This leads to Algorithm 3 described below.

Algorithm 3 Proposed one-versus-one multi-task learning algorithm.

Input: Training samples $X = [X_1, \ldots, X_k]$ with $X_i = [X_i^{(1)}, \ldots, X_i^{(m)}]$, $X_i^{(j)} \in \mathbb{R}^{p \times n_{ij}}$ and test data $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for Task i.
Center and normalize data per task: for all $i' \in \{1, \ldots, k\}$,

- $\mathring{X}_{i'} \leftarrow X_{i'} \left(I_{n_{i'}} - \frac{1}{n_{i'}} \mathbb{1}_{n_{i'}} \mathbb{1}_{n_{i'}}^{\mathsf{T}} \right)$
- $\mathring{X}_{i'} \leftarrow \mathring{X}_{i'} / \frac{1}{n_{i'} p} \operatorname{tr}(\mathring{X}_{i'} \mathring{X}_{i'}^{\mathsf{T}})$.

for $j = 1$ **to** m **do**
  for $j' \in \{1, \ldots, m\} \setminus \{j'\}$ **do**
    Estimate: $\mathbb{M}^{\mathsf{T}} \tilde{Q}_0 \mathbb{M}$ and $\mathscr{V}_{ij}$ according to Remark 10.
    Create optimal scores $\mathscr{y}^{\text{bin}\star}(j', j)$ according to Remark 9.
    Compute classification scores according to (2) and deduce the predicted class $c(j, j') = j$ or $c(j, j') = j'$ based on the decision rule in (6).
  end for
end for
Output: $\hat{j} = \underset{j', j \in \{1, \ldots, m\}}{\text{mode}} \{c(j, j')\}$.[6]

In order to derive the asymptotic correct classification of class ℓ based on Algorithm 3, it is necessary to enumerate all scenarios which lead to the prediction of the class ℓ. Although this could be done in theory, the combinatorics, already for three classes, are cumbersome and not worth developing here. For the specific one-versus-one setting, we therefore do not provide a theoretical performance analysis.

5.4 One-hot encoding approach

For a given Task i in a one-hot encoding approach, using the canonical vector encoding for each class (i.e., $\mathscr{Y}_{ij} = e_j^{[m]}$ encodes all training input data $x_{il}^{(j)}$ of class $\mathscr{C}_j$), the class allocated to an unknown test sample $\mathbf{x}$ is the index of the output vector $g_i(\mathbf{x}) \in \mathbb{R}^m$ with maximum value.

We disrupt here from this approach by explicitly not imposing a one-hot encoding for $\mathscr{Y}_{ij}$. Instead we consider a generic encoding $\mathscr{Y} \in \mathbb{R}^{km \times m}$, which will be optimized in such a way to maximize the classification accuracy.

6. The mode of a set of indices is defined as the most frequent value. When multiple indices occur equally frequently, the smallest of those indices is considered by convention.

Proposition 13 *Under a "one-hot encoding" scheme with generic $\mathscr{Y}$, the probability of correct classification $P_i^{(j)}(\mathbf{x})$ for Task i of a test data $\mathbf{x} \in \mathscr{C}_j$ is given by*

$$P_i^{(j)}(\mathbf{x}) = \underbrace{\int \cdots \int_0^\infty}_{m-1} \frac{1}{\sqrt{(2\pi)^{m-1}|C(j)|}} \exp\left(-\frac{1}{2}(x-\mu(j))^\mathsf{T} C(j)^{-1}(x-\mu(j))\right) dx,$$

where $\mu(j) = \mathscr{E}_j \mathscr{Y}^T \left(I_{mk} - \mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}} \Gamma \mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}}\right) e_{(m(i-1)+j)}^{[km]} \in \mathbb{R}^{m-1}$ *and* $C(j) = \mathscr{E}_j \mathscr{Y}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}} \Gamma \mathscr{V}_{ij} \Gamma \mathscr{D}_{\boldsymbol{\delta}}^{\frac{1}{2}} \mathscr{Y} \mathscr{E}_j^\mathsf{T} \in \mathbb{R}^{(m-1)\times(m-1)}$ *with* $\mathscr{E}_j = \{(e_j^{(m)} - e_{j'}^{(m)})^\mathsf{T}\}_{j \neq j'} \in \mathbb{R}^{(m-1)\times m}$.

A natural objective is to set $\mathscr{Y}$ so to maximize the average correct classification accuracy $\frac{1}{m}\sum_{j=1}^m P_i^{(j)}(\mathbf{x})$ (under assumed uniform prior on $\mathbf{x}$). This form again is not convex in $\mathscr{Y}$ but may be approximated by gradient descent starting from the one-hot encoding solution, as described in Algorithm 4.

Algorithm 4 Proposed "one-hot encoding" multi-task learning algorithm.

Input: Training samples $X = [X_1, \ldots, X_k]$ with $X_i = [X_i^{(1)}, \ldots, X_i^{(m)}]$, $X_i^j \in \mathbb{R}^{p \times n_{ij}}$ and test data $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for target Task i.
Center and normalize data per task: for all $i' \in \{1, \ldots, k\}$,

- $\mathring{X}_{i'} \leftarrow X_{i'} \left(I_{n_{i'}} - \frac{1}{n_{i'}} \mathbb{1}_{n_{i'}} \mathbb{1}_{n_{i'}}^\mathsf{T}\right)$
- $\mathring{X}_{i'} \leftarrow \mathring{X}_{i'} / \frac{1}{n_{i'} p} \mathrm{tr}(\mathring{X}_{i'} \mathring{X}_{i'}^\mathsf{T})$.

Estimate Matrix $\mathbb{M}^\mathsf{T} \tilde{Q}_0 \mathbb{M}$ and $\mathscr{V}_{ij}$ according to Remark 10.
Compute the theoretical score $\mu(j)$ and covariance $C(j)$ and derive the asymptotic classification accuracy $P_i(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^m P_i^{(j)}(\mathbf{x})$.
Create optimal scores $\mathscr{Y}^\star = \arg\max_{\mathscr{Y}} P_i(\mathbf{x})$.
Compute classification scores $g_i(\mathbf{x})$ according to (2).
Output: $\hat{\ell} = \arg\max_{\ell \in \{1,\ldots,m\}} g_i(\mathbf{x}; \ell)$.

6. Experiments

This section has a double objective. The first part (Section 6.1) devises numerical experiments on binary classification settings to corroborate the theoretical analyses and conclusions drawn in this previous section. Here, the target is threefold: (i) empirically illustrate the effects of the bias in the threshold decision and in the label optimization scheme discussed in Section 3, (ii) discuss the impact of numerous tasks ($k > 2$) in the binary class setting, thereby emphasizing the effects of negative transfer and its correction through input score (label) optimization, and (iii) exemplify the relevance of our theoretical findings to a specific application to hypothesis testing in a multi-task setting.

In a second part (Section 6.2), experiments on both synthetic and real data for multi-class classification are realized, which demonstrate, even for real data: (i) the extreme accuracy of the theoretical performance predictions of Propositions 12–13 against empirical data and (ii) the large performance gains induced by the various improvements introduced at length in Section 5.

6.1 Experiments on binary classification

6.1.1 Effect of input score (label) choice

In the present experiment, the effects of the bias in the decision threshold (in general not centered on zero) and of the input score (label) optimization are demonstrated on both synthetic data and real data.

Specifically, MTL-LSSVM is first applied to the following two-task ($k = 2$) and two-class ($m = 2$) setting: for Task 1, $x_{1l}^{(j)} \sim \mathcal{N}((-1)^j \mu_1, I_p)$ (evenly distributed in both classes) and for Task 2, $x_{2l}^{(j)} \sim \mathcal{N}((-1)^j \mu_2, I_p)$ (evenly distributed in both classes), 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 2 depicts the distribution of the binary output scores $g_i^{\text{bin}}(\mathbf{x})$ both for the classical MTL-LSSVM (top displays) and for our proposed random matrix improved scheme, with optimized input labels (bottom display).

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. In effect, the entries of $\mathcal{y}^{\text{bin}\star}$ naturally drop to zero for all unrelated tasks and classes, thereby discarding the undesired use of the latter; the classical binary input labels instead inappropriately exploit these data and induce a negative learning effect, sometimes to such an extent to completely switch the final decision (as here when $\beta = -1$).

For experiments on real data, the MNIST datasets (Deng, 2012) is considered. Specifically, the setting is that of a binary classification for two tasks, mimicking a transfer learning setting: there, the "target" Task 2 aims to discriminate Class $\mathcal{C}_1$ and Class $\mathcal{C}_2$ respectively composed of images of digit 1 and digit 4. The "source" Task 1 is here used as a support for classification in the target task, and consists of the classification of other pairs of digits: either $(5, 9)$, $(9, 5)$, $(6, 2)$ or $(8, 3)$ (we recall that the order of the set of digits (X, Y) is important for the non-optimized MTL-LSSVM since source and target tasks labels are "paired"; thus $(5, 9)$ or $(9, 5)$ digits for the source task will bring different results). We compare here again the non-optimized MTL-LSSVM with labels $\mathcal{y}^{\text{bin}} = [-1, 1, -1, 1]^\mathsf{T}$ to our proposed optimized scheme (as detailed in Remark 9). For both methods, the optimal theoretical threshold decision ζ is used (rather than $\zeta = 0$ for the non-optimized setup) in order to emphasize the influence of input score (label) optimization.

Figure 3 depicts the performance for both methods as a function of the hyperparameter λ. We recall that, as $\lambda \to 0$, the multi-task scheme becomes equivalent to independent

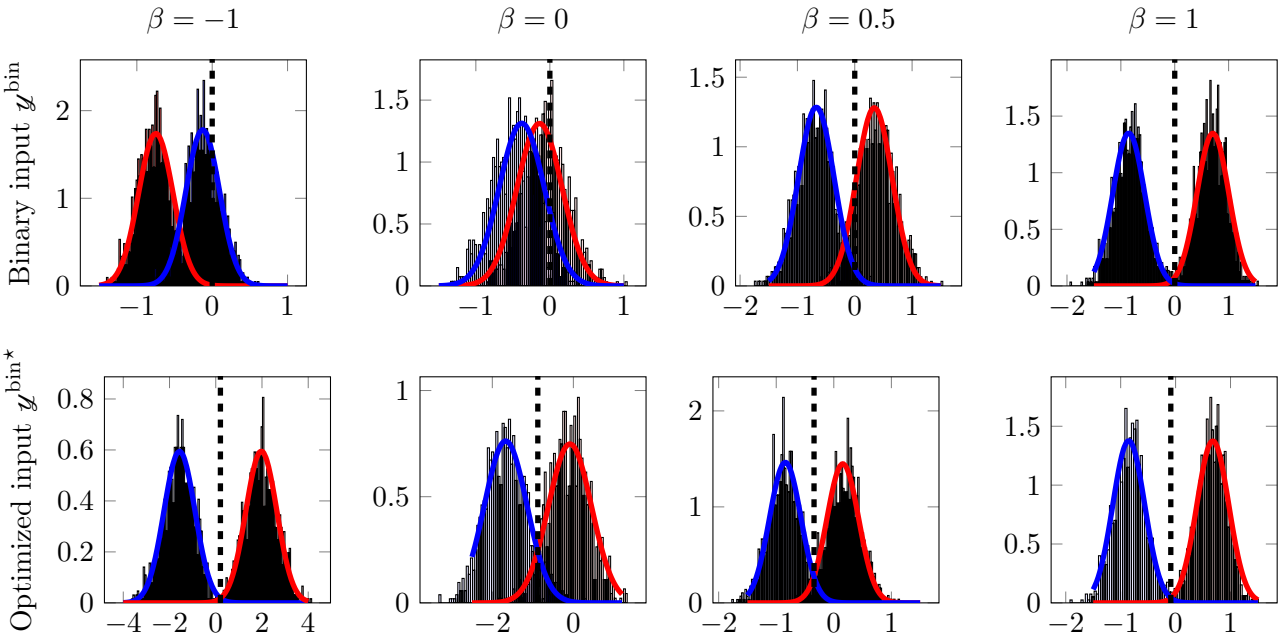

Figure 2: Score distribution for new datum $\mathbf{x}$ of Class $\mathscr{C}_1$ (red) and Class $\mathscr{C}_2$ (blue) for Task 2 in a 2-task ($k = 2$) and 2 class-per-task ($m = 2$) setting of isotropic Gaussian mixtures for: **(top)** classical MTL-LSSVM with no optimization and a threshold assumed at $\zeta = 0$; **(bottom)** proposed optimized MTL-LSSVM with estimated threshold ζ; decision threholds ζ represented in dashed vertical lines; differently related tasks ($\beta = 0$ for orthogonal means, $\beta > 0$ for positively correlated means and $\beta < 0$ for negatively correlated means), $p = 100$, $[c_{11}, c_{12}, c_{21}, c_{22}] = [0.3, 0.4, 0.1, 0.2]$, $\gamma = \mathbb{1}_2$, $\lambda = 10$. Histograms drawn from 1 000 test samples of each class. The figure clearly depicts the deviation from 0 of the decision threshold in unbalanced classes and the deleterious effect of "negative transfer" when β is small; these problems are well handled by the proposed optimized scheme.

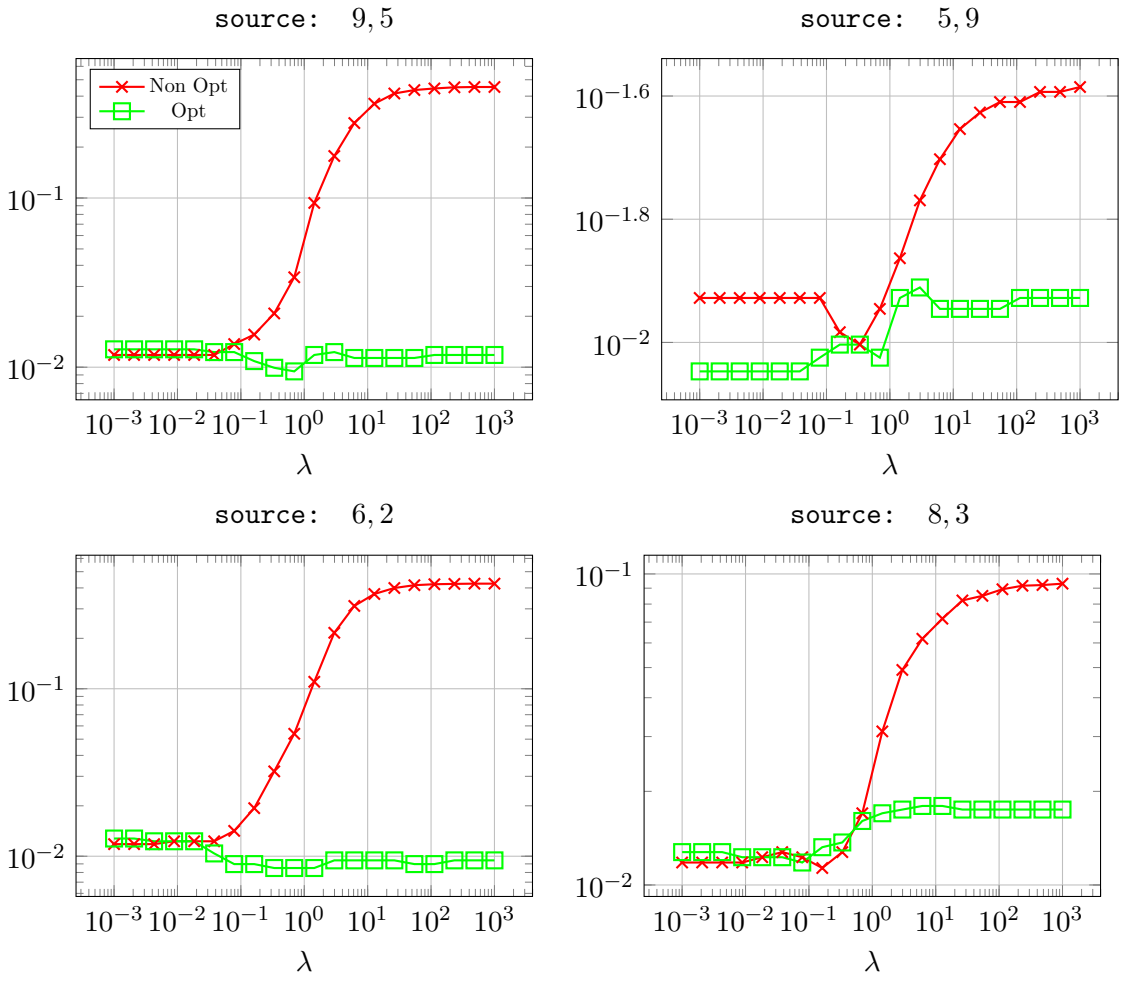

Figure 3: Classification error of digit pair $(1, 4)$ with different source training pairs for classical LSSVM and optimized LSSVM. $n_{11} = n_{12} = 100$, $n_{21} = n_{22} = 10$ and $\gamma = \mathbb{1}_2$. A PCA preprocessing is performed on each image to extract their $p = 100$ principal components; the accuracy is performed over $n_{\text{test}} = 1\,135$ test samples. The proposed method shows a low sensitivity to λ.

single-task classifiers, while as $\lambda \to \infty$, both source and target tasks are considered together as one task. Figure 3 raises the stability of optimal input labelling with respect to λ: this is explained by the fact that $y^{\text{bin}\star}$ is a *function of* λ and thus adapts to each value of λ, even if suboptimal. Besides, for appropriate values of λ, the proposed improved labelling can largely outperform the non-optimized setting, even here on real data.

Table 2 complements the figure by effectively displaying the optimal vectors $y^{\text{bin}\star}$ at the optimal value for λ. The table demonstrates the appropriate adjustment of the labels to the data correlation $\frac{\Delta\mu_1^{\mathsf{T}}\Delta\mu_2}{\|\Delta\mu_2\|^2}$. Specifically, for a negative correlation between the classes of both tasks, the method naturally "switches" the labels (the input data scores) by opposing

[Source]	(9,5)	(5,9)	(6,2)	(8,3)
$\frac{\Delta\mu_1^\mathsf{T}\Delta\mu_2}{\|\Delta\mu_2\|^2}$	-0.2450	0.2450	-0.1670	-0.0818
$y^{\text{bin}\star} = \begin{bmatrix} y_{11}^{\text{bin}\star} \\ y_{12}^{\text{bin}\star} \\ y_{21}^{\text{bin}\star} \\ y_{22}^{\text{bin}\star} \end{bmatrix}$	$\begin{bmatrix} -0.2808 \\ 0.2808 \\ 0.6489 \\ -0.6489 \end{bmatrix}$	$\begin{bmatrix} 0.2808 \\ -0.2808 \\ 0.6489 \\ -0.6489 \end{bmatrix}$	$\begin{bmatrix} -0.2879 \\ 0.2879 \\ 0.6459 \\ -0.6459 \end{bmatrix}$	$\begin{bmatrix} -0.0400 \\ 0.0400 \\ 0.7060 \\ -0.7060 \end{bmatrix}$

Table 2: Optimal input label $y^{\text{bin}\star}$ as a function of the source data pair in the (λ-optimal) configuration of Figure 3.

the signs of $y^{\text{bin}\star}$ in entries $1, 3$ (Class $\mathscr{C}_1$ in each task) and $2, 4$ (Class $\mathscr{C}_2$ in each task). For rather orthogonal tasks (here typically $(8, 3)$), the entries of $y^{\text{bin}\star}$ corresponding to the source task (entries 1 and 2) are almost zero, thereby discarding the source data and avoiding negative transfer. It is also interesting to note that, for moderately correlated tasks (here for the source digits $(5, 9)$), despite the fact that the source task offers ten times more data ($n_{1j} = 100$, $n_{2j} = 10$) and is thus deemed trustworthy for classification, the corresponding entries 1 and 2 in $y^{\text{bin}\star}$ are much smaller than the entries 3, 4 corresponding to the target task: the algorithm thus judges the few target data more relevant to target classification than the many related source tasks.

6.1.2 Analysis of increasing number of tasks

This next experiment illustrates the effect of adding more tasks for the transfer learning setting on synthetic and MNIST datasets. For synthetic data, Gaussian classes with mean $\mu_{ij} = \beta\mu_{i1} + \sqrt{1-\beta^2}\mu_{i1}^\perp$ and various values of β are successively added. For the MNIST dataset, different classifications of digits are added progressively to help classify the specific pair of digits $(1, 4)$. Figure 4 depicts the classification error after each new task addition, both for a classical binary input label choice and for the proposed optimized input labels. The figure forcefully illustrates that our proposed framework avoids negative transfer, as the classification error of MTL never increases as the number of tasks grows. This is quite unlike the non-optimized scheme which severely suffers from negative transfer.

6.1.3 Hypothesis testing

The next experiments, both synthetic and on real data, apply the results of MTL-LSSVM to a hypothesis test on a *target* Task t based on training samples both from a source Task s and the target Task t. For data $\mathbf{x}$ in the target task, the test

$$g_t^{\text{bin}}(\mathbf{x}) \underset{\mathscr{H}_0}{\overset{\mathscr{H}_1}{\gtrless}} \zeta$$

is performed, where $\mathscr{H}_0$ is the null hypothesis (say, Class 2) and $\mathscr{H}_1$ the alternative (say, Class 1) and $\zeta = \zeta(\eta)$ is a decision threshold here selected in such a way to enforce the false alarm rate constraint $P(g_t(\mathbf{x}) \geq \zeta(\eta) \mid \mathbf{x} \in \mathscr{H}_0) \leq \eta$, for a given $\eta \in (0, 1)$. The objective is then to maximize over the input scores y^{bin} the correct detection rate $P(g_t^{\text{bin}}(\mathbf{x}) \geq \zeta(\eta) \mid \mathbf{x} \in$

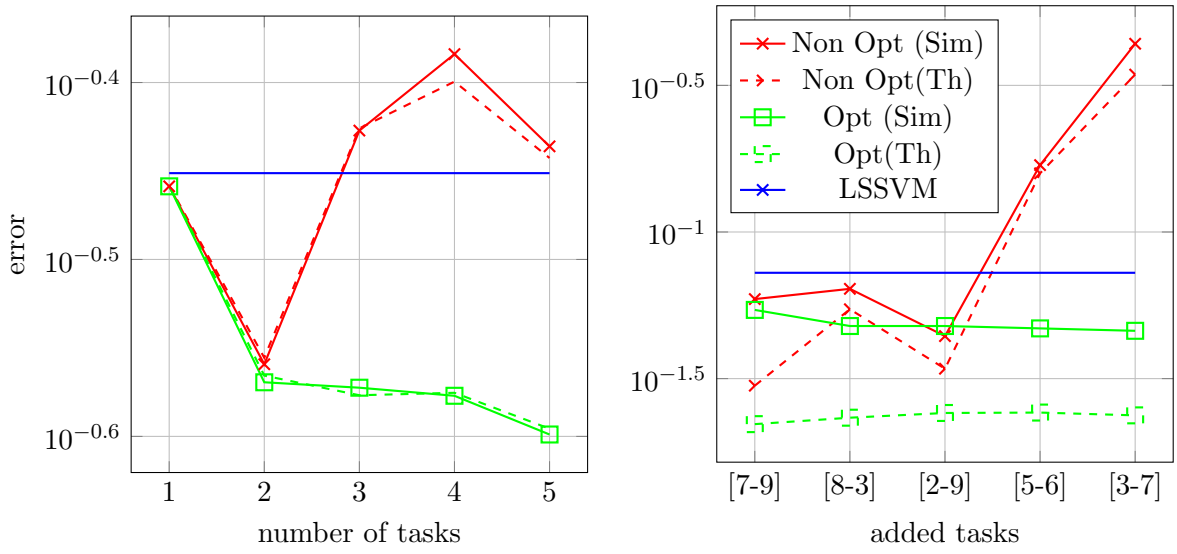

Figure 4: Classification accuracy for increasing number of tasks. **(Left)** Synthetic data with task correlations $\beta = 1, .9, .5, .2, .8$ in this order, $p = 100$ and $c = [.07, .11, .10, .10, .06, .08, .09, .12, .10, .11, .03, .03]^\mathsf{T}$; accuracy evaluated out of 10 000 test samples. **(Right)** MNIST dataset with digits $(1, 4)$ as target task, each added task being shown in x-axis; 100 training samples are used for each class of the source tasks and 10 training samples for each class of the target class; HOG features with $p = 144$ for each image digit; accuracy evaluated out of $n_{\text{test}} = 1\,135$ test samples. For both setting, $\gamma = \mathbb{1}_k$ and $\lambda = 10$. The optimized scheme avoids negative transfer by systematically benefiting from additional tasks.

$\mathscr{H}_1$): this induces a different value for the optimal scores $\mathscr{y}^{\text{bin}\star}$ than proposed in (12), which can be constructed following Remark 9.

The experimental synthetic data is here a two-task ($k = 2$) setting in which $x_{1j} \sim \mathcal{N}(\pm\mu_{11}, I_p)$ (i.e., $\mu_{12} = -\mu_{11}$) and $x_{2j} \sim \mathcal{N}(\pm\mu_{21}, I_p)$, where $\mu_{21} = \beta\mu_{11} + \sqrt{1-\beta^2}\mu_{11}^\perp$, μ_{11} is a unit-norm vector and $\mu_{11}^\perp$ any unit-norm vector orthogonal to μ_{11}. We take here $\beta = 0.5$, so that both tasks are "slightly" correlated. As for the real-world experiment, they are based on the MIT-BIH Arrhythmia dataset (Moody and Mark, 2001). The dataset consists of 109 446 samples from 5 medical heart condition categories: "Normal (N)": 0, "Atrial premature (S)": 1, "Ventricular (V)": 2, "Ventricular-Norma (F)": 3, and "Unclassifiable (Q)": 4. For illustration, we consider here a binary classification with source Classes $\{1, 2\}$ and target Classes $\{3, 4\}$. A false alarm is raised when misclassifying (target) Class 3 into Class 4 and the performance objective consists in maximizing the correct classification of target Class 4.

Figure 5 depicts the algorithm performance through a receiver-operating curve (ROC) for false alarm rates η on both synthetic and real-world data. Both theoretical (Th) asymptotics (used to set the decision threshold ζ) and actual performances (Sim) are displayed, for the optimal (Opt) choice of $\mathscr{y}^{\text{bin}}$ (Opt) and for $\mathscr{y}^{\text{bin}} = [-1, 1, -1, 1]^\mathsf{T}$ (Non-Opt).

Both synthetic and real data graphs of Figure 5 confirm, here under the hypothesis testing problem, the large superiority of our proposed optimized MTL-LSSVM over the standard non-optimized alternative. Besides, the theoretical classification error prediction is an accurate fit to the actual empirical performance, even for not so large values of p and the n_{ij}'s, and even for small error values.[7] This remark is here all the more fundamental that, in practice, η can be set a priori, using Theorem 8 with no need for heavy, unreliable, and data-consuming cross-validation procedures.

6.2 Experiments on multi-class classification

We here consider the complete setting of a $k \geq 2$, $m > 2$ multi-class learning scenario, first on synthetic and then on real image datasets.

6.2.1 Experiments on synthetic dataset

In the synthetic data experiment, the scenario is a two-task ($k = 2$) setting in which $x_{1l}^{(j)} \sim \mathcal{N}(\mu_{1j}, I_p)$ and $x_{2l}^{(j)} \sim \mathcal{N}(\mu_{2j}, I_p)$, where $\mu_{2j} = \beta\mu_{1j} + \sqrt{1-\beta^2}\mu_{1j}^\perp$, with $\mu_{1j} = 2e_j^{[p]}$ and $\mu_{1j}^\perp = e_{p-j}^{[p]}$, and β varies from 0.1 to 0.8.

Table 3 provides the empirical classification accuracy achieved by one-versus-all (Algorithm 2), one-versus-one (Algorithm 3) and one-hot (Algorithm 4) learning versus their standard (non-optimized) algorithm equivalent on 10 000 test samples. The table also reports the theoretical classification accuracies predicted by the empirical estimation of the quantities involved in Propositions 12–13 (therefore without any cross validation) for the one-versus-all and one-hot methods.

7. Since our main result (Theorem 8) is a central limit theorem, it is not expected to be particularly accurate in the "tails" of the distribution of the output scores $g_i^{\text{bin}}(\mathbf{x})$; as such, the observed high accuracy for small error values is remarkable.

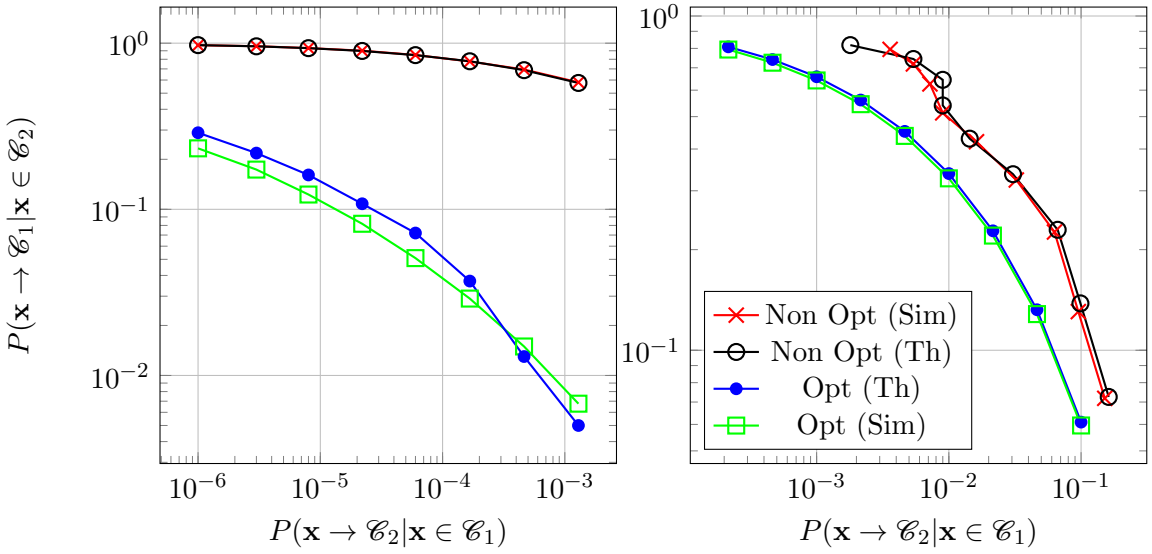

Figure 5: ROC curve for proposed optimized versus standard MTL-LSSVM. **(Left)** Synthetic data with $p = 128$, $n_{11} = 384$, $n_{12} = 256$, $n_{21} = 64$, $n_{22} = 40$, $\mu_{11} = -\mu_{12} = [1, 0, \ldots, 0]^\mathsf{T}$, $\mu_{21} = -\mu_{22} = [.87, .5, 0, \ldots, 0]^\mathsf{T}$. **(Right)** MIT-BIH arrhythmia database, with $p = 550$, $n_{ij} = 500$, $\lambda = 1$, $\gamma = \mathbb{1}_2$. The accuracy of the theoretical anticipation is remarkable and allows for a precise setting of the decision threshold ensuring a desired false alarm rate.

Table 3: Classification accuracy for synthetic data $x_{1l}^{(j)} \sim \mathcal{N}(\mu_{1j}, I_p)$ and $x_{2l}^{(j)} \sim \mathcal{N}(\mu_{2j}, I_p)$, $\mu_{2j} = \beta\mu_{1j} + \sqrt{1-\beta^2}\mu_{1j}^{\perp}$, for different values of the data-correlation $\beta > 0$ and various multi-class learning algorithms. Theoretical performance predictions are provided in parentheses. Here $m = 5$, $p = 100$, $c_{1j} = .16$, $c_{2j} = .04$, for $j \in \{1, \ldots, 5\}$, $\lambda = 1$ and $\gamma = \mathbb{1}_k$. The performance gains of the proposed optimal scheme is particularly clear in tasks with low correlation.

β	Method	one-vs-all	one-vs-one	one-hot
$\beta = 0.1$	Classical	61.43 (59.87)	65.31	65.61 (64.35)
	Optimized	67.63 (67.57)	74.98	67.63 (67.55)
$\beta = 0.5$	Classical	65.47 (66.00)	71.30	67.41 (67.90)
	Optimized	68.00 (68.52)	76.31	68.03 (68.48)
$\beta = 0.8$	Classical	71.16 (70.63)	78.20	70.97 (70.58)
	Optimized	71.19 (70.76)	78.55	71.14 (70.67)

The output performance scores naturally show an improvement using the proposed MTL-LSSVM framework and confirm again the extremely accurate prediction of performance by the theoretical formulas. Most importantly, the table reveals that the gap between the non-optimized and optimized schemes is all the more important that the correlation between task (through the parameter β) is small; this indicates that the optimized MTL-LSSVM learning framework better exploits the (even little) correlation arising between tasks or, alternatively, that the non-optimized scheme suffers from negative learning when "over-emphasizing" the weight of data from the other task (through the binary input labels $\mathcal{Y}$).

As for the comparison of the three classification methods (one-versus-all, one-versus-one and one-hot), it shows here an overall superiority of the one-versus-one approach. This result should nonetheless be interpreted with extreme care as no optimization over the hyperparameters γ, λ is conducted in any scenario.

6.2.2 Image classification

Similarly as in Section 3, we now turn to the popular Office+Caltech256 multi-task image classification benchmark (Saenko et al., 2010; Griffin et al., 2007) often exploited for transfer learning. The overall database consists of 10 categories shared by both Office and Caltech256 datasets. As in Table 1, we consider in sequence the transfer learning of one out of four possible source tasks, each of which consisting in classifying data from one sub-database (images issued from the Caltech set (c), Webcam images (w), Amazon pictures (a) or dslr images (d)), towards another task; this boils down to $4 \times (4-1) = 12$ source-target comparison pairs.)

The results in Table 1 using VGG features for the image representations are extremely close to 100%, already for the "naive" approach consisting in a simplified one-versus-all extension of Algorithm 1. Little would be gained (at least not in computational efforts) by running the more involved Algorithm 2 on the same database. For this reason, for the

present experiment, we compare the more challenging (since less discriminative) $p = 800$ SURF-BoW features of the Office+Caltech256 images instead of their VGG features.

Half of the samples of the target task is randomly selected as test data and the accuracy is evaluated over 20 independent trials. For complexity reasons, as in Section 3, for each experiment, the naive version of the one-versus-all algorithm is run 10 times, considering a fictitious two-class $\tilde{\mathscr{C}}_1$-versus-$\tilde{\mathscr{C}}_2$ setting where, for the classifier focusing on class $\mathscr{C}_\ell$, class $\tilde{\mathscr{C}}_1 = \mathscr{C}_\ell$ while class $\tilde{\mathscr{C}}_2$ is the union of all other classes $\mathscr{C}_{\ell'}$, $\ell' \neq \ell$.

Table 4 reports the accuracy obtained by the algorithm (Proposed) versus the non optimized MTL-LSSVM from (Xu et al., 2013) (LSSVM) and state-of-the-art transfer learning algorithms already introduced in Section 3. Table 4 again demonstrates that our proposed improved MTL-LSSVM, despite its simplicity and unlike the competing methods used for comparison, has stable performances and is highly competitive.

Table 4: Classification accuracy for transfer learning on the Office+Caltech256 database, against state-of-the-art alternatives. Here with c(Caltech), w(Webcam), a(Amazon), d(dslr) based on SURF-BoW features. Our proposed approach is systematically best or second to best and best on average.

S/T	c→ w	w→ c	c→ a	a→ c	w→ a	a→ d	d→ a	w→ d	c→ d	d→ c	a→ w	d→ w	Mean score
LSSVM	79.47	47.70	68.10	49.65	68.13	57.50	70.00	73.75	67.50	46.45	74.83	84.11	65.60
MMDT	69.47	42.55	68.95	39.70	65.24	59.50	62.16	**86.06**	56.94	27.92	68.54	**87.88**	61.24
ILS	24.5	20.92	25.21	21.10	22.92	26.25	27.08	43.75	30.00	26.95	15.23	57.62	28.46
CDLS	*82.28*	**54.21**	*73.75*	**54.49**	**71.52**	*68.56*	*70.54*	69.44	*69.44*	**53.86**	**81.59**	82.78	*69.37*
Ours	**86.09**	*49.65*	**75.00**	*50.35*	*68.83*	**73.75**	**71.25**	*72.50*	**77.50**	*48.05*	*80.13*	*85.43*	**69.88**

7. Concluding remarks

Through the example of multi-task learning, as well as its particularization to transfer learning, the article demonstrates the ability of random matrix theory to predict the performance of advanced machine learning schemes (here based on an extension of LSSVM) and most importantly to propose improved learning mechanisms, which are competitive with, if not largely outperforming, elaborate state-of-the-art alternatives.

Interestingly, as already reported in recent works (Mai and Couillet, 2018; Mai et al., 2019), the proposed random-matrix-optimized framework is largely counter-intuitive and comes along with novel insights on the overall learning mechanisms of large dimensional data classification. Here specifically, the proposed input score (label) optimization is at odds with the conventional binary input label insights of most machine learning schemes, but is key to optimize the exploitation of other tasks and to discard altogether the long standing problem of negative transfer.

The random-matrix framework also draws a significant advantage in its being *universal* to data distributions. As shown here, our main results (Theorem 8) are valid for data modelled as mixtures of concentrated random vectors which go quite beyond the usually assumed

Gaussian mixtures, as they encompass extremely realistic synthetic data models (such as GAN images). This universality phenomenon, possible surprising at first, in fact holds for a wide range of large dimensional "dense" (as opposed to sparse) data representation vectors, encompassing not only images but also likely other forms of data representations, such as word embeddings in natural language processing, vectors of moments of graphons in statistical graph analysis, etc.

To conclude, we importantly emphasize a fundamental underlying take-away message of the present work: recalling that LSSVM is nothing but an explicit and computationally-cheap linear regression method, the fact that it competes or even outperforms elaborate MTL methods testifies of the possibility, when dealing with large dimensional data, to design highly performing elementary and cost-efficient random-matrix-based learning schemes. This remark is in line with the recent parallel analysis of information theoretic bounds on the performances of machine learning problems, such as in (Lelarge and Miolane, 2019) for semi-supervised learning (SSL); similar to the present work, in (Mai and Couillet, 2018), the authors propose a random-matrix-based optimization of standard graph SSL learning which they demonstrate to tightly reach the information theoretic upper bound of (Lelarge and Miolane, 2019). This simultaneously (i) opens the path to a tentative exploration of information-theoretic bounds on transfer learning and multi-task learning for large dimensional data, the results of which could then be confronted to the present proposed scheme, and (ii) strongly suggests the practical relevance of "reinvesting" research efforts in simple, cost-efficient, theoretically tractable, controllable, and usually more stable machine learning schemes, rather than in complex and theoretically intractable techniques.

Acknowledgments

We thank Cosme Louart for fruitful discussions about technical aspects related to concentrated random vectors. This work is supported by the UGA IDEX GSTATS Chair and the MIAI LargeDATA Chair at University Grenoble Alpes.

Appendix A.

A.1 Solution of MTL-LSSVM

The Lagrangian of the constrained optimization problem using the relatedness assumption ($W_i = W_0 + V_i$) reads:

$$\mathscr{L}(\omega_0, v_i, \xi_i, \alpha_i, b_i) = \frac{1}{2\lambda}\mathrm{tr}\left(W_0^\mathsf{T} W_0\right) + \frac{1}{2}\sum_{i=1}^{k}\frac{\mathrm{tr}\left(V_i^\mathsf{T} V_i\right)}{\gamma_i} + \frac{1}{2}\sum_{i=1}^{k}\mathrm{tr}\left(\xi_i^\mathsf{T}\xi_i\right) + \sum_{i=1}^{k}\mathrm{tr}\left(\alpha_i^\mathsf{T}\left(Y_i - \frac{\mathring{X}_i^\mathsf{T} W_0}{kp} - \frac{\mathring{X}_i^\mathsf{T} V_i}{kp} - \mathbb{1}_{n_i} b_i^\mathsf{T} - \xi_i\right)\right)$$

with $\alpha_i \in \mathbb{R}^{n_i \times m}$ the Lagrangian parameter attached to task i.

Differentiating with respect to the unknowns W_0, V_i, ξ_i, α_i, and b_i leads to the following system of equations:

$$\frac{1}{\lambda}W_0 - \sum_{i=1}^{k} X_i\alpha_i = 0 \tag{14}$$

$$\frac{1}{\gamma_i}V_i - X_i\alpha_i = 0 \tag{15}$$

$$\xi_i - \alpha_i = 0 \tag{16}$$

$$Y_i - \frac{\mathring{X}_i^\mathsf{T} W_0}{kp} - \frac{\mathring{X}_i^\mathsf{T} V_i}{kp} - \mathbb{1}_{n_i} b_i^\mathsf{T} - \xi_i = 0 \tag{17}$$

$$\alpha_i^\mathsf{T}\mathbb{1}_{n_i} = 0. \tag{18}$$

Plugging the expression of W_0 (Equation (14)), V_i (Equation (15)) and ξ_i (Equation (16)) into Equation (17) leads to:

$$Y_i = (\lambda + \gamma_i)\frac{\mathring{X}_i^\mathsf{T}\mathring{X}_i}{kp}\alpha_i + \lambda\sum_{j\neq i}\frac{\mathring{X}_i^\mathsf{T} X_j}{kp}\alpha_j + \mathbb{1}_{n_i} b_i^\mathsf{T} + \alpha_i$$

$$\mathbb{1}_{n_i}^\mathsf{T}\alpha_i = 0.$$

With $Y = [Y_1^\mathsf{T}, \ldots, Y_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^n$, $\alpha = [\alpha_1^\mathsf{T}, \ldots, \alpha_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^n$, $Z = \sum_{i=1}^{k} e_i^{[k]} e_i^{[k]\mathsf{T}} \otimes \mathring{X}_i \in \mathbb{R}^{kp\times n}$ and $P \in \mathbb{R}^{n\times k}$ such that the j-th column is $P_{\cdot j} = [\mathbf{0}_{n_1+\ldots+n_{j-1}}^\mathsf{T}, \mathbb{1}_{n_j}^\mathsf{T}, \mathbf{0}_{n_{j+1}+\ldots+n_k}^\mathsf{T}]^\mathsf{T}$, this system of equations can be written under the following compact matrix form:

$$Pb + Q^{-1}\alpha = Y$$

$$P^\mathsf{T}\alpha = \mathbf{0}_k$$

with $Q = \left(\frac{Z^\mathsf{T} A Z}{kp} + I_n\right)^{-1} \in \mathbb{R}^{n\times n}$, and $A = \left(\mathscr{D}_\gamma + \lambda\mathbb{1}_k\mathbb{1}_k^\mathsf{T}\right) \otimes I_p \in \mathbb{R}^{kp\times kp}$.

Solving for α and b then gives:

$$\alpha = Q(Y - Pb)$$

$$b = (P^\mathsf{T} Q P)^{-1} P^\mathsf{T} Q Y.$$

Moreover, using $W_i = W_0 + V_i$ and Equations (14) and (15), the expression of W_i becomes:

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

A.2 Calculus of deterministic equivalents

Lemma 14 (Deterministic equivalents) *Define, for class $\mathscr{C}_j$ in Task i, the data deterministic matrices*

$$M = \left(e_1^{[k]} \otimes [\mu_{11}, \ldots, \mu_{1m}], \ldots, e_k^{[k]} \otimes [\mu_{k1}, \ldots, \mu_{km}] \right)$$
$$\mathbb{C}_{ij} = A^{\frac{1}{2}} \left(e_i^{[k]} {e_i^{[k]}}^\mathsf{T} \otimes (\Sigma_{ij} + \mu_{ij}\mu_{ij}^\mathsf{T}) \right) A^{\frac{1}{2}}.$$

Then we have the deterministic equivalents of first order

$$\tilde{Q} \leftrightarrow \bar{\tilde{Q}} \equiv \left(\sum_{i=1}^{k} \sum_{j=1}^{m} \boldsymbol{\delta}_{ij}^{[mk]} \mathbb{C}_{ij} + I_{kp} \right)^{-1}$$
$$A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z \leftrightarrow A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T}$$

and of second order

$$\tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} \leftrightarrow B_{ij}$$
$$Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z \leftrightarrow J M_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} (B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T} - \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij}) + F_{ij}$$

in which we defined

$$W_{ij} = [w_{11}, \ldots, w_{km}]^\mathsf{T}, \quad w_{sl} = \left[\boldsymbol{0}_{n_{11}+\ldots+n_{(s-1)l}}^\mathsf{T}, \frac{2\boldsymbol{\delta}_{sl}^{[mk]} \mathrm{tr}\left(B_{ij}\mathbb{C}_{sl}\right)}{n_{sl}} \mathbb{1}_{n_{sl}}^\mathsf{T}, \boldsymbol{0}_{n_{(s+1)l}+\ldots+n_{km}}^\mathsf{T} \right]^\mathsf{T}$$

$$F_{ij} = \sum_{i',j'} \frac{c_0^2 {\boldsymbol{\delta}_{i'j'}^{[mk]}}^2}{c_{i'j'}^2} \mathrm{tr}(\mathbb{C}_{i'j'} B_{ij}) e_{i'j'}^{[mk]} {e_{i'j'}^{[mk]}}^\mathsf{T}$$

$$B_{ij} = \bar{\tilde{Q}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \bar{\tilde{Q}} + \sum_{i'=1}^{k} \sum_{j'=1}^{2} d_{i'j'} T_{ij,i'j'} [\bar{\tilde{Q}} \mathbb{C}_{i'j'} \bar{\tilde{Q}}]$$

$$D = \sum_{i,j} d_{ij} e_{ij}^{[mk]} {e_{ij}^{[mk]}}^\mathsf{T}, \; d_{ij} = \frac{c_0}{c_{ij}} {\boldsymbol{\delta}_{ij}^{[mk]}}^2$$

$$J = [j_{11}, \ldots, j_{km}],$$

$$j_{lm} = \left(0_{n_{11}+\ldots+n_{(i-1)m}}^\mathsf{T}, \mathbb{1}_{n_{ij}}^\mathsf{T}, 0_{n_{(i+1)1}+\ldots+n_{km}}^\mathsf{T} \right)^\mathsf{T},$$

$$M_{\boldsymbol{\delta}} = M \sum_{ij} \frac{c_0}{c_{ij}} \boldsymbol{\delta}_{ij}^{[mk]} e_{ij}^{[mk]} {e_{ij}^{[mk]}}^\mathsf{T}$$

$$S_{ij} = e_i^{[k]} {e_i^{[k]}}^\mathsf{T} \otimes \Sigma_{ij}$$

$$T = \bar{T}(I_k - D\mathscr{T})^{-1}, \; \mathscr{T}_{ij,i'j'} = \frac{1}{kp} \mathrm{tr}(\mathbb{C}_{ij} \bar{\tilde{Q}} \mathbb{C}_{i'j'} \bar{\tilde{Q}}), \bar{T}_{ij,i'j'} = \frac{1}{kp} tr\left(\mathbb{C}_{i'j'} \bar{\tilde{Q}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \bar{\tilde{Q}} \right)$$

and the $(\boldsymbol{\delta}_{11}^{[mk]}, \ldots, \boldsymbol{\delta}_{km}^{[mk]})$ *are the unique positive solutions of*

$$\boldsymbol{\delta}_{ij}^{[mk]} = \frac{c_{ij}}{c_0 \left(1 + \frac{1}{kp}\mathrm{tr}(\mathbb{C}_{ij}\bar{\tilde{Q}})\right)}, \ \forall i, j.$$

A.2.1 Proof of Lemma 14

First order deterministic equivalent. A deterministic equivalent for $\tilde{Q}$ is retrieved similarly as provided in (Louart and Couillet, 2018). Our objective is then to find, based on this result, a deterministic equivalent for the random matrix $A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}Z$. To this end, we evaluate the scalar quantity $\mathbb{E}[u^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}Zv]$ for any deterministic vector $u \in \mathbb{R}^{kp}$ and $v \in \mathbb{R}^n$ such that $\|u\| = 1$ and $\|v\| = 1$, which we can write

$$\mathbb{E}\left[u^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}Zv\right] = \sum_{i=1}^{n} v_i \mathbb{E}\left[u^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}z_i\right]. \tag{19}$$

Furthermore, let us define for convenience the matrix Z_{-i}, which is the matrix Z with a vector of $\mathbf{0}$ on its i-th column such that $ZZ^\mathsf{T} = Z_{-i}Z_{-i}^\mathsf{T} + z_i z_i^\mathsf{T}$. Using the Sherman-Morrison matrix inversion lemma (i.e., $(A + uv^\mathsf{T})^{-1} = A^{-1} - \frac{A^{-1}uv^\mathsf{T}A^{-1}}{1+v^\mathsf{T}A^{-1}u}$), we find:

$$\tilde{Q} = \left(\frac{A^{\frac{1}{2}}ZZ^\mathsf{T}A^{\frac{1}{2}}}{kp} + I_{kp}\right)^{-1} = \tilde{Q}_{-i} - \frac{1}{kp}\frac{\tilde{Q}_{-i}A^{\frac{1}{2}}z_i z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}}{1 + \frac{1}{kp}z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i} \tag{20}$$

with $\tilde{Q}_{-i} = (\frac{A^{\frac{1}{2}}Z_{-i}Z_{-i}^\mathsf{T}A^{\frac{1}{2}}}{kp} + I_{kp})^{-1}$. Furthermore,

$$\tilde{Q}A^{\frac{1}{2}}z_i = \frac{\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}{1 + \frac{1}{kp}z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}. \tag{21}$$

Plugging Equation (21) into Equation (19) leads to

$$\mathbb{E}\left[u^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}A^{\frac{1}{2}}Zv\right] = \sum_{i=1}^{n} v_i \mathbb{E}\left[u^\mathsf{T}\frac{A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}{1 + \frac{1}{kp}z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}\right]. \tag{22}$$

Moreover, following the same line of reasoning as in (Seddik et al., 2020, Proposition A.3), based on Assumption 1 and tools from concentration of measure theory (see also (Ledoux, 2001; Louart et al., 2018)), one can show that:

$$\sum_{i=1}^{n} v_i \mathbb{E}\left[u^\mathsf{T}\frac{A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}{1 + \frac{1}{kp}z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}\right] = \sum_{i=1}^{n} v_i \mathbb{E}\left[u^\mathsf{T}\frac{A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i}{1 + \boldsymbol{\delta}_{ij}}\right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \tag{23}$$

with $\boldsymbol{\delta}_{ij} \equiv \mathbb{E}\left[\frac{1}{kp}z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_i\right]$. Note that $\boldsymbol{\delta}_{ij}$ can be estimated as the solution of the fixed point equation

$$\boldsymbol{\delta}_{ij} = \frac{1}{kp}\mathbb{E}\left[\mathrm{tr}\left(A^{\frac{1}{2}}z_i z_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}\right)\right] = \frac{1}{kp}\mathrm{tr}\left(\mathbb{C}_{ij}\bar{\tilde{Q}}\right) + \mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$$

since z_i's are independent from $\tilde{Q}_{-i}$.

We then conclude that:

$$\mathbb{E}\left[u^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z v\right] = \sum_{i=1}^{n} v_i u^\mathsf{T} \frac{\mathbb{E}\left[A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} z_i\right]}{1+\boldsymbol{\delta}_{ij}} + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) = u^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} v + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)$$

where in the last equality, we used the fact that $\tilde{Q}_{-i}$ is independent from z_i. This concludes the proof.

Second order deterministic equivalent We aim in the following section to prove that $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z \leftrightarrow J M_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} (B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T} - \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij}) + F_{ij}$.

Let us define for convenience $\mathscr{C}(i)$ the class of the i-th sample. Similarly as done for the first order deterministic equivalents, the focus will be on $\mathbb{E}[u^\mathsf{T} Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z v]$.

In order to obtain an estimate of this bilinear form, or equivalently here a deterministic equivalent for $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z$, one must isolate the contribution the off-diagonal versus diagonal elements of the latter matrix. Starting with the off-diagonal elements, using

successively Equation (20) and Equation (23) on i and j , we have

$$
\begin{aligned}
&\sum_{\substack{i,j=1\\ i\neq j}}^{n} u_i v_i \mathbb{E}\left[z_i^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} z_j\right] \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{n} u_i v_i \mathbb{E}\left[\frac{z_i^\mathsf{T} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-j} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})}\right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{n} u_i v_i \mathbb{E}\left[\frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})} - \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \tilde{Q}_{-j} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})}\right. \\
&\left. - \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} z_j z_j^\mathsf{T} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-j} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})}\right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{n} u_i v_i \mathbb{E}\left[\frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})} - \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(i)})}\right. \\
&- \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} z_j z_j^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})} \\
&\left. + \frac{1}{(kp)^2} \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} z_j z_j^\mathsf{T} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(i)})}\right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right) \\
&= \sum_{\substack{i,j=1\\ i\neq j}}^{n} u_i v_i \mathbb{E}\left[\frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})} - \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(i)})}\right. \\
&\left. - \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} z_j z_j^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{kp(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})}\right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right).
\end{aligned}
$$

where the term

$$
\frac{1}{(kp)^2} \frac{z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-j}{\tilde{Q}_{-i}} A^{\frac{1}{2}} z_j z_j^\mathsf{T} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \underset{-i}{\tilde{Q}_{-j}} A^{\frac{1}{2}} z_j}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})(1+\boldsymbol{\delta}_{\mathscr{C}(j)})(1+\boldsymbol{\delta}_{\mathscr{C}(i)})}
$$

is proved to be order $\mathcal{O}(\frac{1}{\sqrt{p}})$ using (Seddik et al., 2020, Lemma A.2).

As such, the "sub-deterministic equivalent" for the matrix $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z$ with diagonal elements discarded is $JM_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T} - JM_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij}$, with

$$A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} \leftrightarrow B_{ij}$$
$$W_{ij} = [w_{11}, \ldots, w_{km}]^\mathsf{T}, \quad w_{sl} = \left[\mathbf{0}_{n_{11}+\ldots+n_{(s-1)l}}^\mathsf{T}, \frac{2\mathrm{tr}\left(B_{ij}\mathbb{C}_{sl}\right)}{kp(1+\boldsymbol{\delta}_{sl})} \mathbb{1}_{n_{sl}}^\mathsf{T}, \mathbf{0}_{n_{(s+1)l}+\ldots+n_{km}}^\mathsf{T} \right]^\mathsf{T}$$

(note that this matrix estimator of the off-diagonal elements is not zero on the diagonal; however its diagonal elements vanish as $n, p \to \infty$ and may thus be maintained without affecting the final result).

We next need to handle the contribution of the diagonal elements. These are obtained similarly as the off-diagonal elements and lead to the deterministic diagonal matrix equivalent

$$F_{ij} = \sum_{i',j'} \frac{\mathrm{tr}(\mathbb{C}_{i'j'} B_{ij})}{(1+\boldsymbol{\delta}_{i'j'})^2} e_{i'j'}^{[mk]} e_{i'j'}^{[mk]\mathsf{T}}.$$

Put together, the complete deterministic equivalent is then:

$$JM_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T} - JM_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij} + \sum_{i',j'} \frac{\mathrm{tr}(\mathbb{C}_{i'j'} B_{ij})}{(1+\boldsymbol{\delta}_{i'j'})^2} e_{i'j'}^{[mk]} e_{i'j'}^{[mk]\mathsf{T}}.$$

This proves that $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z \leftrightarrow JM_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} (B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T} - \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij}) + F_{ij}$.

Calculus of B_{ij}. To conclude the proof of Lemma 14, it then remains to find a deterministic equivalent for $\tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}$ which we denote by B_{ij}. Similar derivations and results are provided in detail in (Louart et al., 2018). For conciseness, we sketch the most important elements of the proof. The interested reader can refer to (Louart et al., 2018, Section 5.2.3). Let us evaluate $\mathbb{E}[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} (\tilde{Q} - \bar{\tilde{Q}}) v]$ for any deterministic vector $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$ such that $\|u\| = 1$ and $\|v\| = 1$ by using successively Equations (23) and (20):

$$\begin{aligned}
\mathbb{E}\left[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} (\tilde{Q} - \bar{\tilde{Q}}) v \right] &= \mathbb{E}\left[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} (-\frac{A^{\frac{1}{2}} Z Z^\mathsf{T} A^{\frac{1}{2}}}{kp} + \mathbb{C}_{\boldsymbol{\delta}}) \bar{\tilde{Q}} v \right] \\
&= -\frac{1}{kp} \sum_i \mathbb{E}\left[\frac{u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} v}{1+\boldsymbol{\delta}_{ij}} \right] \\
&\quad + \mathbb{E}\left[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-i} \mathbb{C}_{\boldsymbol{\delta}} \bar{\tilde{Q}} v \right] \\
&\quad - \frac{1}{kp} \mathbb{E}\left[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} \mathbb{C}_{\boldsymbol{\delta}} \bar{\tilde{Q}} v \right] + \mathcal{O}\left(\sqrt{\frac{\log p}{p}} \right)
\end{aligned}$$

where $\mathbb{C}_{\boldsymbol{\delta}} = \sum_{ij} \frac{c_{ij}}{c_0} \frac{\mathbb{C}_{ij}}{1+\boldsymbol{\delta}_{ij}}$. Using Assumption 1 and following the work of Louart and Couillet (2018),

$$\frac{1}{kp} \mathbb{E}\left[u^\mathsf{T} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q}_{-i} A^{\frac{1}{2}} z_i z_i^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} \mathbb{C}_{\boldsymbol{\delta}} \bar{\tilde{Q}} v \right] = \mathcal{O}(\frac{1}{p}).$$

Furthermore,

$$\begin{aligned}\mathbb{E}\left[u^\mathsf{T}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}(\tilde{Q}-\bar{\tilde{Q}})v\right] &= -\frac{1}{kp}\sum_i \mathbb{E}\left[\frac{u^\mathsf{T}\tilde{Q}_{-i}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_iz_i^\mathsf{T}A^{\frac{1}{2}}\bar{\tilde{Q}}v}{1+\boldsymbol{\delta}_{ij}}\right]\\ &\quad+\frac{1}{kp}\sum_i \mathbb{E}\left[\frac{u^\mathsf{T}\tilde{Q}_{-i}A^{\frac{1}{2}}z_iz_i^\mathsf{T}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}_{-i}A^{\frac{1}{2}}z_iz_i^\mathsf{T}A^{\frac{1}{2}}\bar{\tilde{Q}}v}{kp(1+\boldsymbol{\delta}_{ij})^2}\right]\\ &\quad+\mathbb{E}\left[u^\mathsf{T}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}Q_{-i}\mathbb{C}_{\boldsymbol{\delta}}\bar{\tilde{Q}}v\right]+\mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)\\ &= \frac{1}{kp}\sum_i \mathbb{E}\frac{\mathrm{tr}\left(\mathbb{C}_{\mathscr{C}(i)}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}\right)}{(1+\boldsymbol{\delta}_{\mathscr{C}(i)})^2}\mathbb{E}\left[u^\mathsf{T}\bar{\tilde{Q}}\mathbb{C}_{\mathscr{C}(i)}\bar{\tilde{Q}}v\right]+\mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)\end{aligned}$$

where $-\frac{1}{kp}\sum_i \mathbb{E}[\frac{u^\mathsf{T}\tilde{Q}_{-i}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}_{-i}z_iz_i^\mathsf{T}\bar{\tilde{Q}}v}{1+\boldsymbol{\delta}_{ij}}]+\mathbb{E}[u^\mathsf{T}\tilde{Q}_{-i}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}Q\mathbb{C}_{\boldsymbol{\delta}}\bar{\tilde{Q}}v]=\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$, following again (Louart and Couillet, 2018).

Let us next denote $d_{ab}=\frac{n_{ab}}{kp(1+\boldsymbol{\delta}_{ab})^2}$. We then have the following identity for $\mathbb{E}[\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}]$:

$$\mathbb{E}[\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}]=\bar{\tilde{Q}}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\bar{\tilde{Q}}+\sum_{i'=1}^k\sum_{j'=1}^m\frac{d_{i'j'}}{kp}\mathbb{E}\left[\mathrm{tr}\left(\mathbb{C}_{i'j'}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}\right)\right]\bar{\tilde{Q}}\mathbb{C}_{i'j'}\bar{\tilde{Q}}+\mathcal{O}_{\|\cdot\|}\left(\sqrt{\frac{\log p}{p}}\right)\tag{24}$$

Further introduce the two matrices $\bar{T}$ and T defined as: $\bar{T}_{ab,ij}=\frac{1}{kp}\mathrm{tr}(\mathbb{C}_{ab}\bar{\tilde{Q}}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\bar{\tilde{Q}})$ and $T_{ij,i'j'}=\frac{1}{kp}\mathbb{E}[\mathrm{tr}\left(\mathbb{C}_{i'j'}\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}\right)]$. These satisfy the following equations (i.e., by right multiplying Equation (24) by $\mathbb{C}_{i'j'}$ and taking the trace)

$$T^{(ij)}_{i'j'}=\bar{T}_{ij,i'j'}+\sum_{e=1}^k\sum_{f=1}^m d_{ef}T_{ef,ij}\mathscr{T}_{i'j',ef},$$

so that $T=\bar{T}(I_k-D\mathscr{T})^{-1}$ where $D=\mathscr{D}_{[d_{11},\ldots,d_{km}]^\mathsf{T}}$ and $\mathscr{T}_{ef,i'j'}=\frac{1}{kp}\mathrm{tr}(\mathbb{C}_{ef}\bar{\tilde{Q}}\mathbb{C}_{i'j'}\bar{\tilde{Q}})$.

Finally,

$$\tilde{Q}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\tilde{Q}\leftrightarrow\bar{\tilde{Q}}A^{\frac{1}{2}}S_{ij}A^{\frac{1}{2}}\bar{\tilde{Q}}+\sum_{i'=1}^k\sum_{j'=1}^m d_{i'j'}T^{(ij)}_{i'j'}\mathbb{E}[\bar{\tilde{Q}}\mathbb{C}_{i'j'}\bar{\tilde{Q}}]\tag{25}$$

with $T=\bar{T}(I_k-D\mathscr{T})^{-1}$.

A.3 Proof of Theorem 8

Proof of the convergence in distribution. Under a Gaussian mixture assumption for the input data X, the convergence in distribution of the statistics of the classification score $g_i(\mathbf{x})$ is identical to the central limit theorem derived in (Liao and Couillet, 2019, Appendix B) by writing the classification score $g_i(\mathbf{x})$ in polynomial form of a Gaussian vector and by resorting to the Lyapounov central limit theorem (Billingsley, 2008).

Since conditionally on the training data X, the classification score $g(x)$ is expressed as the projection of the deterministic vector W on the concentrated random vector $\mathbf{x}$, the CLT for concentrated vector unfolds by proving that projections of deterministic vector on concentrated random vector is asymptotically gaussian. This is ensured by the following result.

Theorem 15 (CLT for concentrated vector (Klartag, 2007; Fleury et al., 2007)) *If $\mathbf{x}$ is a concentrated random vector with $\mathbb{E}[\mathbf{x}] = 0$, $\mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}] = I_p$ with an observable diameter of order $\mathcal{O}(1)$ and σ be the uniform measure on the sphere $\mathcal{S}^{p-1} \subset \mathbb{R}^p$ of radius 1, then for any integer k, small compared to p, there exist two constants C, c and a set $\Theta \subset (\mathcal{S}^{p-1})^k$ such that $\underbrace{\sigma \otimes \ldots \otimes \sigma}_{k}(\Theta) \geq 1 - \sqrt{p}Ce^{-c\sqrt{p}}$ and $\forall \theta = (\theta_1, \ldots, \theta_k) \in \Theta$,*

$$\forall a \in \mathbb{R}^k : \sup_{t \in \mathbb{R}} |\mathbb{P}(a^\mathsf{T}\theta^\mathsf{T}\mathbf{x} \geq t) - G(t)| \leq Cp^{-\frac{1}{4}}.$$

with $G(t)$ the cumulative distribution function of $\mathcal{N}(0,1)$

Then the result unfolds naturally.

Statistical mean of the classification scores. Using the definition of the score in (2), the average output score $g_i(\mathbf{x})$ for $\mathbf{x} \in \mathscr{C}_j$ is

$$\mathbb{E}[g_i(\mathbf{x})] = \mathbb{E}\left[\frac{1}{kp}\left(e_i^{[k]} \otimes \mu_{ij}\right)^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z(Y - Pb)\right] + b_i.$$

Using Lemma 14, this can be further developed as:

$$\mathbb{E}[g_i(\mathbf{x})] = \frac{1}{kp}\left(e_i^{[k]} \otimes \mu_{ij}\right)^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J^\mathsf{T}(Y - P\bar{b}) + b_i + o(1). \tag{26}$$

Since $\mathbb{C}_{ij} = A^{\frac{1}{2}}(e_i^{[k]} e_i^{[k]\mathsf{T}} \otimes (\Sigma_{ij} + \mu_{ij}\mu_{ij}^\mathsf{T}))A^{\frac{1}{2}}$ is a finite rank update of Σ_{ij}, one can further use Woodbury identity matrix (i.e., $(A + UCV)^{-1} = A^{-1} + A^{-1}UC(I + VA^{-1}U)VA^{-1}$ for invertible square A) to write $\bar{\tilde{Q}} = \bar{\tilde{Q}}_0 - \bar{\tilde{Q}}_0 \mathbb{M}(I_{kp} + \mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0\mathbb{M})^{-1}\mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0$, with

$$\bar{\tilde{Q}}_0 = \left[\sum_{i=1}^{k}\sum_{j=1}^{m} (\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k)^{\frac{1}{2}} e_i e_i^\mathsf{T} (\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k)^{\frac{1}{2}} \otimes \boldsymbol{\delta}_{ij}^{[mk]} \Sigma_{ij} + I_{kp}\right]^{-1}$$
$$\mathbb{M} = A^{\frac{1}{2}} M \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}}$$

with $\boldsymbol{\delta}^{[mk]} = [\boldsymbol{\delta}_{i1}^{[mk]}, \ldots, \boldsymbol{\delta}_{mk}^{[mk]}]$ for $\boldsymbol{\delta}_{ij}^{[mk]} = \frac{c_{ij}}{c_0(1+\boldsymbol{\delta}_{ij}^{[mk]})}$. Plugging the expression of $\bar{\tilde{Q}}$ in Equation (26), we obtain

$$\begin{aligned}\mathbb{E}[g_i(\mathbf{x})] &= e_{ij}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{-\frac{1}{2}} \mathbb{M}^\mathsf{T} \bar{\tilde{Q}} \mathbb{M} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} + b_i + o(1) \\ &= e_{ij}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{-\frac{1}{2}} (I_{mk} - \Gamma) \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} + b_i + o(1)\end{aligned}$$

with $\Gamma = (I_{mk} + \mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0\mathbb{M})^{-1}$ and $e_{ij}^{[mk]}$ is the canonical vector. Finally, to be exhaustive without going into the technical details,[8] let us conclude by remarking that one can show using the deterministic equivalent for Q provided in (Louart and Couillet, 2018) that $b_i = \frac{\mathbb{1}_{n_i}^\mathsf{T} Y_i}{n_i} + \mathcal{O}(p^{-\frac{1}{2}}) = \mathring{\mathscr{Y}} - \mathring{\mathscr{Y}} + \mathcal{O}(p^{-\frac{1}{2}})$.

Finally, letting m_{ij} be the above expression of $\mathbb{E}[g_i(\mathbf{x})]$ without the trailing $o(1)$ and $m = [m_{11}, \ldots, m_{km}]^\mathsf{T}$, one concludes using the notations of Theorem 8 that

$$m = \mathscr{Y} - \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{-\frac{1}{2}} \Gamma \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}}$$

as desired.

Variance of the classification score. Using Equation (2), for $\mathbf{x} \in \mathscr{C}_j$, the covariance of the score $g_i(\mathbf{x})$ is given by

$$\mathrm{Cov}[g_i(\mathbf{x})] = \mathbb{E}\left[\frac{1}{(kp)^2}(Y - Pb)^\mathsf{T} Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z (Y - Pb)\right]$$

Using the deterministic equivalent of $Z^\mathsf{T} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} \tilde{Q} A^{\frac{1}{2}} Z$ in Lemma 1, the expression further reads

$$\begin{aligned}\mathrm{Cov}[g_i(\mathbf{x})] = = &\frac{1}{(kp)^2}(Y - P\bar{b})^\mathsf{T} \left(J M_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} B_{ij} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} J + F_{ij}\right)(Y - P\bar{b}) \\ &- \frac{1}{p^2}(Y - P\bar{b})^\mathsf{T} J M_{\boldsymbol{\delta}}^\mathsf{T} A^{\frac{1}{2}} \bar{\tilde{Q}} A^{\frac{1}{2}} M_{\boldsymbol{\delta}} W_{ij} (Y - P\bar{b}).\end{aligned}$$

Similarly to the calculus performed for $\mathbb{E}[g_i(\mathbf{x})]$, using again $\bar{\tilde{Q}} = \bar{\tilde{Q}}_0 - \bar{\tilde{Q}}_0 \mathbb{M}(I_{kp} + \mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0\mathbb{M})^{-1}\mathbb{M}^\mathsf{T}\bar{\tilde{Q}}_0$, similar algebraic manipulations lead to:

$$\begin{aligned}\mathrm{Cov}[g_i(\mathbf{x})] = = &\mathring{\mathscr{Y}}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathbb{M}^\mathsf{T} B_{ij} \mathbb{M} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} + \mathring{\mathscr{Y}}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathscr{D}_{\kappa_{ij,\cdot}} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} - \mathring{\mathscr{Y}}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathbb{M}^\mathsf{T} \bar{\tilde{Q}} \mathbb{M} \mathscr{D}_{\frac{\kappa_{ij,\cdot}}{\boldsymbol{\delta}^{[mk]}}} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} \\ = &\mathscr{Y}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \Gamma \mathbb{M}^\mathsf{T} \bar{\tilde{Q}}_0 \mathbb{V}_{ij} \bar{\tilde{Q}}_0 \mathbb{M} \Gamma \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathscr{Y} + \mathscr{Y}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} (I - \Gamma) \mathscr{D}_{\kappa_{ij,\cdot}} (I - \Gamma) + \\ &\mathring{\mathscr{Y}}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathscr{D}_{\kappa_{ij,\cdot}} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} - 2\mathring{\mathscr{Y}}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} (I - \Gamma) \mathscr{D}_{\kappa_{ij,\cdot}} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathring{\mathscr{Y}} \\ = &\mathscr{Y}^\mathsf{T} \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \left(\Gamma \mathscr{D}_{\kappa_{ij,\cdot}} \Gamma + \Gamma \mathbb{M}^\mathsf{T} \bar{\tilde{Q}}_0 \mathbb{V}_{ij} \bar{\tilde{Q}}_0 \mathbb{M} \Gamma\right) \mathscr{D}_{\boldsymbol{\delta}^{[mk]}}^{\frac{1}{2}} \mathscr{Y}\end{aligned}$$

with $\mathbb{V}_{ij} = A^{\frac{1}{2}} S_{ij} A^{\frac{1}{2}} + \sum_{i'=1}^{k} \sum_{j'=1}^{m} \boldsymbol{\delta}_{i'j'}^{[mk]} \kappa_{ij,i'j'} A^{\frac{1}{2}} S_{i'j'} A^{\frac{1}{2}}$ and $\kappa_{ij,\cdot} = [\kappa_{ij,11}, \ldots, \kappa_{ij,k2}]$ with $\kappa_{ij,i'j'} = d_{i'j'} T_{ij,i'j'} / \boldsymbol{\delta}_{i'j'}^{[mk]}$.

8. Due to Remark 3, b_i can take any arbitrary value since only the decision threshold but not the performance is sensitive to a shift of Y.

A.3.1 Particular Case

In the case of binary classification ($m = 2$) and for $\Sigma_{ij} = I_p$, we have the simplification:

$$\begin{aligned}\mathbb{M} &= \sum_{i,j} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{\frac{1}{2}} e_i^{[k]} e_i^{[k]\mathsf{T}} \otimes \sqrt{\tilde{\boldsymbol{\delta}}_i} \mathring{\mu}_{ij} \\ &= \sum_{i} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{\frac{1}{2}} e_i^{[k]} e_i^{[k]\mathsf{T}} \otimes \left(\frac{\left[\frac{c_{i2}\sqrt{c_{i1}}}{c_i}, -\frac{c_{i1}\sqrt{c_{i2}}}{c_i}\right]}{c_0(1+\boldsymbol{\delta}_i)} \otimes \Delta\mu_i \right).\end{aligned}$$

Moreover, $\bar{\tilde{Q}}_0 = [(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T})^{\frac{1}{2}} \mathscr{D}_{\tilde{\boldsymbol{\delta}}} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{\frac{1}{2}} + I_k]^{-1} \otimes I_p$, so that

$$\begin{aligned}\mathbb{M}^\mathsf{T} \bar{\tilde{Q}}_0 \mathbb{M} &= \sum_{i,j} e_i^{[k]} e_i^{[k]\mathsf{T}} \left[I_k + \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{-1} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \right]^{-1} e_j^{[k]} e_j^{[k]\mathsf{T}} \Delta\mu_i^\mathsf{T} \Delta\mu_j \otimes \mathbb{c}_i \mathbb{c}_j^\mathsf{T} \\ &= \sum_{i,j} \mathscr{A}_{ij} \Delta\mu_i^\mathsf{T} \Delta\mu_j e_i^{[k]} e_j^{[k]\mathsf{T}} \otimes \mathbb{c}_i \mathbb{c}_j^\mathsf{T} \\ &= \left(\mathscr{A} \otimes \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right) \odot \mathscr{M}\end{aligned}$$

with

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

As for the covariance terms,

$$\begin{aligned}\mathbb{M}^\mathsf{T} \bar{\tilde{Q}}_0 V_{ij} \bar{\tilde{Q}}_0 \mathbb{M} = \sum_{i,j} e_i^{[k]} e_i^{[k]\mathsf{T}} &\left[I_k + \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(\mathscr{D}_\gamma + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T}\right)^{-1} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \right]^{-1} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(e_i^{[k]} e_i^{[k]\mathsf{T}} + \mathscr{D}_{\kappa_i \odot \tilde{\boldsymbol{\delta}}} \right) \times \\ &\mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left[I_k + \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(I_k + \lambda \mathbb{1}_k \mathbb{1}_k^\mathsf{T} \right)^{-1} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \right]^{-1} e_j^{[k]} e_j^{[k]\mathsf{T}} \Delta\mu_i^\mathsf{T} \Delta\mu_j \otimes \mathbb{c}_i \mathbb{c}_j^\mathsf{T} \\ &= \sum_{i,j} \left[\mathscr{A} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(e_i^{[k]} e_i^{[k]\mathsf{T}} + \mathscr{D}_{\kappa_i \odot \tilde{\boldsymbol{\delta}}} \right) \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \mathscr{A} \right]_{ij} \Delta\mu_i^\mathsf{T} \Delta\mu_j e_i^{[k]} e_j^{[k]\mathsf{T}} \otimes \mathbb{c}_i \mathbb{c}_j^\mathsf{T} \\ &= \left(\mathscr{A} \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \left(e_i^{[k]} e_i^{[k]\mathsf{T}} + \mathscr{D}_{\kappa_i \odot \tilde{\boldsymbol{\delta}}} \right) \mathscr{D}_{\tilde{\boldsymbol{\delta}}}^{-\frac{1}{2}} \mathscr{A} \otimes \mathbb{1}_k \mathbb{1}_k^\mathsf{T} \right) \odot \mathscr{M} \\ &= \frac{1}{\boldsymbol{\delta}_i^{[k]}} \left(\mathscr{A} \mathscr{D}_{\mathscr{K}_i + e_i^{[k]}} \mathscr{A} \otimes \mathbb{1}_k \mathbb{1}_k^\mathsf{T} \right) \odot \mathscr{M}\end{aligned}$$

with $\mathscr{K}_{ia} = \tilde{\boldsymbol{\delta}}_i \kappa_{ia}$. Using Equation (25), after algebraic manipulations, we finally obtain the compact form

$$\mathscr{K} = \frac{c_0}{k} [\mathscr{A} \odot \mathscr{A}] \left(\mathscr{D}_c - \frac{c_0}{k} [\mathscr{A} \odot \mathscr{A}] \right)^{-1}. \tag{27}$$

A.4 Proof of Propositions 12–13

A.4.1 One-versus-all

The probability of correct classification for Task i and for a test data $\mathbf{x} \in \mathscr{C}_j$ reads

$$\mathbb{P}\left(g_i^{\text{bin}}(\mathbf{x}; j) > \max_{j' \neq j}\{g_i^{\text{bin}}(\mathbf{x}; j')\}\right) = \mathbb{P}\left(g_i^{\text{bin}}(\mathbf{x}; j) - \max_{j' \neq j}\{g_i^{\text{bin}}(\mathbf{x}; j')\} > 0\right).$$

Since by definition (Equation (2))

$$g_i^{\text{bin}}(\mathbf{x}; j) = \frac{1}{kp}\mathring{\mathscr{y}}(j)^\mathsf{T} J^\mathsf{T} Q Z^\mathsf{T} A\left(e_i^{[k]} \otimes \mathring{\mathbf{x}}\right) + b_i, \tag{28}$$

we have that $g_i^{\text{bin}}(\mathbf{x}, j)\mathbf{1}_{m-1} - \left\{g_i^{\text{bin}}(\mathbf{x}; j')\right\}_{j' \neq j} = \frac{1}{kp}\mathscr{Y}_{-j} J^\mathsf{T} Q Z^\mathsf{T} A\left(e_k^{[k]} \otimes \mathring{\mathbf{x}}\right)$, where $\mathscr{Y}_{-j} = (\mathring{\mathscr{y}}(j)^\mathsf{T} - \left[\mathring{\mathscr{y}}(j')^\mathsf{T}\right]_{j' \neq j}) \in \mathbb{R}^{(m-1)\times km}$. Using Theorem 8 with $\mathscr{Y}$ replaced by $\mathscr{Y}_{-j}$, $g_i^{\text{bin}}(\mathbf{x}, j)\mathbf{1}_{m-1} - g_i^{\text{bin}}(\mathbf{x}; j')_{j' \neq j} \in \mathbb{R}^{m-1}$ is asymptotically a multivariate Gaussian random vector with statistics detailed in the theorem statement. Proposition 12 then unfolds trivially by remarking that $g_i^{\text{bin}}(\mathbf{x}; j) > \max_{j' \neq j} g_i^{\text{bin}}(\mathbf{x}; j') \Leftrightarrow \forall j' \neq j,\ g_i^{\text{bin}}(\mathbf{x}; j) - g_i^{\text{bin}}(\mathbf{x}; j') \geq 0$.

A.4.2 One Hot encoding

The proof is similar to the one-versus-all case.

The probability of correct classification for a test data $\mathbf{x} \in \mathscr{C}_j$ is

$$\mathbb{P}\left(g_i^{\text{bin}}(\mathbf{x}; j) > \max_{j' \neq j}\{g_i^{\text{bin}}(\mathbf{x}; j')\}\right) = \mathbb{P}\left(g_i^{\text{bin}}(\mathbf{x}; j) - \max_{j' \neq j}\{g_i^{\text{bin}}(\mathbf{x}; j')\} > 0\right)$$

where

$$g_i^{\text{bin}}(\mathbf{x}; j) = \frac{1}{kp} {e_j^{[k]}}^\mathsf{T} \mathring{\mathscr{Y}}^\mathsf{T} J^\mathsf{T} Q Z^\mathsf{T} A\left(e_i^{[k]} \otimes \mathring{\mathbf{x}}\right) + b_i. \tag{29}$$

Therefore $g_i^{\text{bin}}(\mathbf{x}; j)\mathbf{1}_{m-1} - \{(g_i(\mathbf{x}; j'))\}_{j' \neq j} = \frac{1}{kp}\mathscr{E}_j \mathring{\mathscr{Y}}^\mathsf{T} J^\mathsf{T} Q Z^\mathsf{T} A(e_k^{[k]} \otimes \mathring{\mathbf{x}})$, with $\mathscr{E}_j = \{(e_j^{(m)} - e_{j'}^{(m)})^\mathsf{T}\}_{j \neq j'} \in \mathbb{R}^{(m-1)\times m}$. By Theorem 8 with $\mathring{\mathscr{Y}}$ replaced by $\mathscr{E}_j^\mathsf{T}\mathring{\mathscr{Y}}^\mathsf{T}$, this vector is asymptotically normally distributed and Proposition 13 unfolds immediately using again the fact that $g_i^{\text{bin}}(\mathbf{x}; j) > \max_{j' \neq j} g_i^{\text{bin}}(\mathbf{x}; j') \Leftrightarrow \forall j' \neq j,\ g_i^{\text{bin}}(\mathbf{x}; j) - g_i^{\text{bin}}(\mathbf{x}; j'j) \geq 0$.

References

Arvind Agarwal, Samuel Gerber, and Hal Daume. Learning multiple tasks using manifold regularization. In *Advances in neural information processing systems*, pages 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*, pages 41–48, 2007.

Z. Bai and J. W. Silverstein. Spectral Analysis of Large Dimensional Random Matrices. *Springer Series in Statistics*, 2009.

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*, pages 567–580. Springer, 2003.

Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Rich Caruana, Shumeet Baluja, and Tom Mitchell. Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in neural information processing systems*, pages 959–965, 1996.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

R. Couillet and M. Debbah. *Random matrix methods for wireless communications*. Cambridge University Press, New York, NY, USA, first edition, 2011.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

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

Bruno Fleury, Olivier Guédon, and Grigoris Paouris. A stability result for mean width of lp-centroid bodies. *Advances in Mathematics*, 214(2):865–877, 2007.

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*, pages 2672–2680, 2014.

William H Greene. Econometric analysis 4th edition. *International edition, New Jersey: Prentice Hall*, pages 201–215, 2000.

G Griffin, A Holub, and P Perona. Caltech-256 object category dataset california inst. Technical report, Technol., Tech. Rep. 7694, 2007 [Online]. Available: http://authors. library ..., 2007.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*, 2017.

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*, pages 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*, pages 5081–5090, 2016.

Bo'az Klartag. A central limit theorem for convex sets. *Inventiones mathematicae*, 168(1): 91–131, 2007.

Sajja Tulasi Krishna and Hemantha Kumar Kalluri. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(5S4):427–432, 2019.

Michel Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical surveys and monographs. American Mathematical Soc., 2001. ISBN 0821837923.

Marc Lelarge and Léo Miolane. Asymptotic bayes risk for gaussian mixture in a semi-supervised setting. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 639–643. IEEE, 2019.

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.

Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *arXiv preprint arXiv:1805.08295*, 2018.

Cosme Louart, Zhenyu Liao, Romain Couillet, et al. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

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 and Zhenyu Liao. High dimensional classification via empirical risk minimization: Improvements and optimality. *arXiv preprint arXiv:1905.13742*, 2019.

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)*, pages 3357–3361. IEEE, 2019.

Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

Leonid Andreevich Pastur and Mariya Shcherbina. *Eigenvalue distribution of large random matrices*. Number 171 in Mathematical Surveys and Monographs. American Mathematical Soc., 2011.

Anderson Rocha and Siome Klein Goldenstein. Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):289–302, 2013.

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

Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. Kernel random matrices of large concentrated data: the example of gan-generated images. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484. IEEE, 2019.

Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. *arXiv preprint arXiv:2001.08370*, 2020.

Vladimir Vapnik. Universal learning technology: Support vector machines. *NEC Journal of Advanced Technology*, 2(2):137–144, 2005.

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.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan): 35–63, 2007.

Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge University Press, 2020.

Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, 2005.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.