PCA-based Multi Task Learning: a Random Matrix Approach

Malik Tiomoko
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
91190, Gif-sur-Yvette, France. malik.tiomoko@u-psud.fr

Romain Couillet
Gipsa Lab
Université Grenoble Alpes
romain.couillet@gipsa-lab.grenoble-inp.fr

Frédéric Pascal
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
91190, Gif-sur-Yvette, France
frederic.pascal@centralesupelec.fr

Abstract

The article proposes and theoretically analyses a *computationally efficient* multi-task learning (MTL) extension of popular principal component analysis (PCA)-based supervised learning schemes [7, 5]. The analysis reveals that (i) by default learning may dramatically fail by suffering from *negative transfer*, but that (ii) simple counter-measures on data labels avert negative transfer and necessarily result in improved performances.

Supporting experiments on synthetic and real data benchmarks show that the proposed method achieves comparable performance with state-of-the-art MTL methods but at a *significantly reduced computational cost*.

1 Introduction

From single to multiple task learning. Advanced supervised machine learning algorithms require large amounts of *labelled* samples to achieve high accuracy, which in practice is often too demanding. Multi-task learning (MTL) [11, 52, 53] and *transfer learning* provide a potent workaround by appending extra *somewhat similar* datasets to the scarce available dataset of interest. The additional data possibly being of a different nature, MTL effectively solves multiple tasks *in parallel* while exploiting task relatedness to enforce collaborative learning.

State-of-the-art of MTL. To proceed, MTL solves multiple related tasks and introduces shared hyperparameters or feature spaces, optimized to improve the performance of the individual tasks. The crux of efficient MTL lies in both enforcing and, most importantly, evaluating task relatedness: this in general is highly non-trivial as this implies to theoretically identify the common features of the datasets. Several heuristics have been proposed which may be split in two groups: parameter- versus feature-based MTL. In parameter-based MTL, the tasks are assumed to share common hyperparameters [15, 49] (*e.g.*, separating hyperplanes in a support vector machine (SVM) flavor) or hyperparameters derived from a common prior distribution [54, 55]. Classical learning mechanisms (SVM, logistic regression, etc.) can be appropriately turned into an MTL version by enforcing

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

parameter relatedness: [15, 49, 35] respectively adapt the SVM, least square-SVM (LS-SVM), and large margin nearest neighbor (LMNN) methods into an MTL paradigm. In feature-based MTL, the data are instead assumed to share a common low-dimensional representation, which needs be identified: through sparse coding, deep neural network embeddings, principal component analysis (PCA) [2, 32, 50, 34] or simply by feature selection [33, 48, 18].

The negative transfer plague. A strong limitation of MTL methods is their lack of theoretical tractability: as a result, the biases inherent to the base methods (SVM, LS-SVM, deep nets) are exacerbated in MTL. A major consequence is that many of these heuristic MTL schemes suffer from *negative transfer*, i.e., cases where MTL performs worse than a single-task approach [40, 29]; this often occurs when task relatedness is weaker than assumed and MTL enforces fictitious similarities.

A large dimensional analysis to improve MTL. Based on a large dimensional random matrix setting, this work focuses on an elementary (yet powerful) PCA-based MTL approach and provides an exact (asymptotic) evaluation of its performance. This analysis conveys insights into the MTL inner workings, which in turn provides an optimal data labelling scheme to fully avert negative transfer.

More fundamentally, the choice of investigating PCA-based MTL results from realizing that the potential gains incurred by a proper theoretical adaptation of simple algorithms largely outweigh the losses incurred by biases and negative transfer in more complex and elaborate methods (see performance tables in the article). As a result, the main contribution of the article lies in achieving *high performance MTL at low computational cost* when compared to competitive methods.

This finding goes in the direction of the compellingly needed development of cost-efficient and environment-friendly AI solutions [24, 44, 20].

Article contributions. In detail, our main contributions may be listed as follows:

- We theoretically compare the performance of two *natural* PCA-based single-task supervised learning schemes (PCA and SPCA) and justify the uniform superiority of SPCA;
- As a consequence, we propose a natural extension of SPCA to multi-task learning for which we also provide an asymptotic performance analysis;
- The latter analysis (i) theoretical grasps the transfer learning mechanism at play, (ii) exhibits the relevant information being transferred, and (iii) harnesses the sources of negative transfer;
- This threefold analysis unfolds in a *counter-intuitive* improvement of SPCA-MTL based on an optimal data label adaptation (not set to ± 1, which is the very source of negative transfer); *the label adaptation depends on the optimization target*, changes from task to task, and can be efficiently computed prior to running the SPCA-MTL algorithm;
- Synthetic and real data experiments support the competitive SPCA-MTL results when compared to state-of-the-art MTL methods; these experiments most crucially show that high performance levels can be achieved at significantly lower computational costs.

Supplementary material. The proofs and Matlab codes to reproduce our main results and simulations, along with theoretical extensions and additional supporting results, are provided in the supplementary material.

Notation. $e_m^{[n]} \in \mathbb{R}^n$ is the canonical vector of $\mathbb{R}^n$ with $[e_m^{[n]}]_i = \delta_{mi}$. Moreover, $e_{ij}^{[mk]} = e_{m(i-1)+j}^{[mk]}$.

2 Related works

A series of supervised (single-task) learning methods were proposed which rely on PCA [7, 39, 51, 16]: the central idea is to project the available data onto a shared low-dimensional space, thus ignoring individual data variations. These algorithms are generically coined supervised principal component analysis (SPCA). Their performances are however difficult to grasp as they require to understand the statistics of the PCA eigenvectors: only recently have large dimensional statistics, and specifically random matrix theory, provided first insights into the behavior of eigenvalues and eigenvectors of sample covariance and kernel matrices [8, 23, 4, 25, 37]. To the best of our knowledge, none of these works have drawn an analysis of SPCA: the closest work is likely [3] which however only provides statistical bounds on performance rather than exact results.

On the MTL side, several methods were proposed under unsupervised [30, 43, 6], semi-supervised [38, 28] and supervised (parameter-based [46, 15, 49, 1] or feature-based [2, 27]) flavors. Although most of these works generally achieve satisfying performances on both synthetic and real data, few theoretical analyses and guarantees exist, so that instances of negative transfer are likely to occur.

To be exhaustive, we must mention that, for specific types of data (images, text, time series) and under the availability of numerous labelled samples, deep learning MTL methods have recently been devised [41]. These are however at odds with the article requirement to leverage scarce labelled samples and to be valid for generic inputs (beyond images or texts): these methods cannot be compared on even grounds with the methods discussed in the present study.[1]

3 Supervised principal component analysis: single task preliminaries

Before delving into PCA-based MTL, first results on large dimensional PCA-based single-task learning for a training set $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n}$ of n samples of dimension p are needed. To each $x_i \in \mathbb{R}^p$ is attached a label y_i: in a binary class setting, $y_i \in \{-1, 1\}$, while for $m \geq 3$ classes, $y_i = e_j^{[m]} \in \mathbb{R}^m$, the canonical vector of the corresponding class j.

PCA in supervised learning. Let us first recall that, applied to X, PCA identifies a subspace of $\mathbb{R}^p$, say the span of the columns of $U = [u_1, \ldots, u_\tau] \in \mathbb{R}^{p \times \tau}$ ($\tau \leq p$), which maximizes the variance of the data when projected on the subspace, i.e., U solves:

$$\max_{U \in \mathbb{R}^{p \times \tau}} \operatorname{tr}\left(U^\mathsf{T} \frac{XX^\mathsf{T}}{p} U \right) \text{ subject to } U^\mathsf{T} U = I_\tau.$$

The solution is the collection of the eigenvectors associated with the τ largest eigenvalues of $\frac{XX^\mathsf{T}}{p}$.

To predict the label $\mathbf{y}$ of a test data vector $\mathbf{x}$, a simple method to exploit PCA consists in projecting $\mathbf{x}$ onto the PCA subspace U and in performing classification in the projected space. This has the strong advantage to provide a (possibly dramatic) dimensionality reduction (from p to τ) to supervised learning mechanisms, thus improving cost efficiency while mitigating the loss incurred by the reduction in dimension. Yet, the PCA step is fully unsupervised and does not exploit the available class information. It is instead proposed in [7, 12] to trade U for a more representative projector V which "maximizes the dependence" between the projected data $V^\mathsf{T} X$ and the output labels $y = [y_1, \ldots, y_n]^\mathsf{T} \in \mathbb{R}^{m \times n}$. To this end, [7] exploits the Hilbert-Schmidt independence criterion [19], with corresponding optimization

$$\max_{V \in \mathbb{R}^{p \times \tau}} \operatorname{tr}\left(V^\mathsf{T} \frac{Xyy^\mathsf{T}X^\mathsf{T}}{np} V \right) \text{ subject to } V^\mathsf{T} V = I_\tau.$$

This results in the *Supervised PCA* (SPCA) projector, the solution $V = V(y)$ of which being the concatenation of the τ dominant eigenvectors of $\frac{Xyy^\mathsf{T}X^\mathsf{T}}{np}$. Subsequent learning (by SVMs, empirical risk minimizers, discriminant analysis, etc.) is then applied to the projected training $V^\mathsf{T} x_i$ and test $V^\mathsf{T}\mathbf{x}$ data. For binary classification where y is unidimensional, $\frac{Xyy^\mathsf{T}X^\mathsf{T}}{np}$ is of rank 1, which reduces $V^\mathsf{T}\mathbf{x}$ to the scalar $V^\mathsf{T}\mathbf{x} = y^\mathsf{T} X^\mathsf{T}\mathbf{x} / \sqrt{y^\mathsf{T} X^\mathsf{T} X y}$, i.e., to a mere matched filter.

Large dimensional analysis of SPCA. To best grasp the performance of PCA- or SPCA-based learning, assume the data arise from a large dimensional m-class Gaussian mixture.[2]

Assumption 1 (Distribution of X) *The columns of X are independent random vectors with $X = [X_1, \ldots, X_m]$, $X_j = [x_1^{(j)}, \ldots, x_{n_j}^{(j)}] \in \mathbb{R}^{p \times n_j}$ for $x_i^{(j)} \sim \mathcal{N}(\mu_j, I_p)$, also denoted $x_i^{(j)} \in \mathcal{C}_j$. We further write $M \equiv [\mu_1, \ldots, \mu_m] \in \mathbb{R}^{p \times m}$.*

[1] But nothing prevents us to exploit data features extracted from pretrained deep nets.

[2] To obtain simpler intuitions, we consider here an *isotropic* Gaussian mixture model (i.e., with identity covariance). This strong constraint is relaxed in the supplementary material, where arbitrary covariances are considered; the results only marginally alter the main conclusions.

Assumption 2 (Growth Rate) *As $n \to \infty$, $p/n \to c_0 > 0$, the feature dimension τ is constant and, for $1 \leq j \leq m$, $n_j/n \to c_j > 0$; we denote $c = [c_1, \ldots, c_m]^\mathsf{T}$ and $\mathcal{D}_c = \mathrm{diag}(c)$. Besides,*

$$(1/c_0)\mathcal{D}_c^{\frac{1}{2}} M^\mathsf{T} M \mathcal{D}_c^{\frac{1}{2}} \to \mathcal{M} \in \mathbb{R}^{m \times m}.$$

We will show that, under this setting, SPCA is uniformly more discriminative on new data than PCA.

As $n, p \to \infty$, the spectrum of $\frac{1}{p} XX^\mathsf{T}$ is subject to a *phase transition phenomenon* now well established in random matrix theory [4, 8]. This result is crucial as the PCA vectors of $\frac{1}{p} XX^\mathsf{T}$ are *only informative* beyond the phase transition and otherwise can be considered as pure noise.

Proposition 1 (Eigenvalue Phase transition) *Under Assumptions 1-2, as $n, p \to \infty$, the* empirical spectral measure $\frac{1}{p}\sum_{i=1}^p \delta_{\lambda_i}$ *of the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_p$ of $\frac{XX^\mathsf{T}}{p}$ converges weakly, with probability one, to the Marčenko-Pastur law [31] supported on $[(1-\sqrt{1/c_0})^2, (1+\sqrt{1/c_0})^2]$. Besides, for $1 \leq i \leq m$, and for $\ell_1 > \ldots > \ell_m$ the eigenvalues of $\mathcal{M}$,*[3]

$$\lambda_i \xrightarrow{\text{a.s.}} \begin{cases} \bar{\lambda}_i \equiv 1 + \frac{1}{c_0} + \ell_i + \frac{1}{c_0 \ell_i} \geq (1+\sqrt{1/c_0})^2 & , \ell_i \geq \frac{1}{\sqrt{c_0}} \\ (1+\sqrt{1/c_0})^2 & , \text{otherwise} \end{cases} ; \quad \lambda_{m+1} \xrightarrow{\text{a.s.}} (1+\sqrt{1/c_0})^2.$$

Proposition 1 states that, if $\ell_i \geq 1/\sqrt{c_0}$, the i-th largest eigenvalue of $\frac{1}{p} XX^\mathsf{T}$ separates from the main *bulk* of eigenvalues. These isolated eigenvalues are key to the proper functioning of PCA-based classification as their corresponding eigenvectors are non-trivially related to the class discriminating statistics (here the μ_j's). Consequently, $U^\mathsf{T}\mathbf{x} \in \mathbb{R}^\tau$ also exhibits a phase transition phenomenon.

Theorem 1 (Asymptotic behavior of PCA projectors) *Let $\mathbf{x} \sim \mathcal{N}(\mu_j, I_p)$ independent of X. Then, under Assumptions 1-2, with $(\ell_i, \bar{u}_i)$ the decreasing (distinct) eigenpairs of $\mathcal{M}$, as $p, n \to \infty$,*

$$U^\mathsf{T}\mathbf{x} - G_j \to 0, \quad G_j \sim \mathcal{N}(\mathfrak{m}_j^{(\mathrm{pca})}, I_\tau), \quad \text{in probability,}$$
$$\text{where } [\mathfrak{m}_j^{(\mathrm{pca})}]_i = \begin{cases} \sqrt{\frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)}}\, \bar{u}_i^\mathsf{T} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} e_j^{[m]} & , i \leq \min(m,\tau) \text{ and } \ell_i \geq \frac{1}{\sqrt{c_0}} \\ 0 & , \text{otherwise.} \end{cases}$$

As such, only the projections on the eigenvectors of $\frac{1}{p} XX^\mathsf{T}$ attached to *isolated* eigenvalues carry informative discriminating features. Practically, for all n, p large, it is thus useless to perform PCA on a larger dimension than the number of isolated eigenvalues, i.e., $\tau \leq \arg\max_{1 \leq i \leq m} \{\ell_i \geq 1/\sqrt{c_0}\}$.

Consider now SPCA. Since $\frac{Xyy^\mathsf{T}X^\mathsf{T}}{np}$ only has m non-zero eigenvalues, no phase transition occurs: all eigenvalues are "isolated". One may thus take $\tau = m$ principal eigenvectors for the SPCA projection matrix V, these eigenvectors being quite likely informative.

Theorem 2 (Asymptotic behavior of SPCA projectors) *Let $\mathbf{x} \sim \mathcal{N}(\mu_j, I_p)$ independent of X. Then, under Assumptions 1-2, as $p, n \to \infty$, in probability,*

$$V^\mathsf{T}\mathbf{x} - G_j \to 0, \quad G_j \sim \mathcal{N}(\mathfrak{m}_j^{(\mathrm{spca})}, I_\tau), \quad [\mathfrak{m}_j^{(\mathrm{spca})}]_i = \sqrt{1/(\tilde{\ell}_i)}\; \bar{v}_i^\mathsf{T} \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} e_j^{[m]}$$

for $\tilde{\ell}_1 \geq \ldots \geq \tilde{\ell}_m$ the eigenvalues of $\mathcal{D}_c + \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{\frac{1}{2}}$ and $\bar{v}_1, \ldots, \bar{v}_m$ their associated eigenvectors.

Since both PCA and SPCA data projections $U^\mathsf{T}\mathbf{x}$ and $V^\mathsf{T}\mathbf{x}$ are asymptotically Gaussian and isotropic (i.e., with identity covariance), the oracle-best supervised learning performance only depends on the differences $\mathfrak{m}_j^{(\times)} - \mathfrak{m}_{j'}^{(\times)}$ ($\times$ being pca or spca). In fact, being small dimensional (of dimension τ), *the vectors $\mathfrak{m}_j^{(\times)}$ can be consistently estimated* from their associated empirical means, and are known in the large n, p limit (with probability one).

[3] We implicitly assume the ℓ_i's distinct for simplicity of exposition.

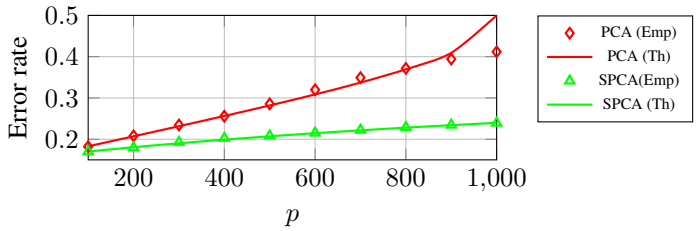


Figure 1: Theoretical (Th) vs. empirical (Emp) error for PCA- and SPCA-based binary classification: $x_i^{(\ell)} \sim \mathcal{N}((-1)^\ell \mu, I_p)$ ($\ell \in \{1,2\}$), $\mu = e_1^{[p]}$, $n_1 = n_2 = 500$. Averaged over 1 000 test samples.

Remark 1 (Consistent estimate of sufficient statistics) *From Assumption 2, c_j can be empirically estimated by n_j/n. This in turns provides a consistent estimate for $\mathcal{D}_c$. Besides, as $n, p \to \infty$,*

$$\mathbb{1}_{n_j}^\mathsf{T} X_j^\mathsf{T} X_{j'} \mathbb{1}_{n_{j'}} \xrightarrow{\text{a.s.}} [M^\mathsf{T} M]_{jj'},\ \forall j \neq j' \quad \textit{and} \quad \mathbb{1}_{\frac{n_j}{2}}^\mathsf{T} X_{j,1}^\mathsf{T} X_{j,2} \mathbb{1}_{\frac{n_j}{2}} \xrightarrow{\text{a.s.}} [M^\mathsf{T} M]_{jj},\ \forall j$$

where $X_j = [X_{j,1}, X_{j,2}] \in \mathbb{R}^{p \times n_j}$, with $X_{j,1}, X_{j,2} \in \mathbb{R}^{p \times (n_j/2)}$. Combining the results provides a consistent estimate for $\mathcal{M}$ as well as an estimate $\hat{\mathfrak{m}}_j^{(\times)}$ for the quantities $\mathfrak{m}_j^{(\times)}$, by replacing c and $\mathcal{M}$ by their respective estimates in the definition of $\mathfrak{m}_j^{(\times)}$.

These results ensure the (large n, p) optimality of the classification decision rule, for a test data $\mathbf{x}$:

$$\underset{j \in \{1,\ldots,m\}}{\arg\max}\ \|U^\mathsf{T}\mathbf{x} - \hat{\mathfrak{m}}_j^{(\text{pca})}\|^2, \qquad \underset{j \in \{1,\ldots,m\}}{\arg\max}\ \|V^\mathsf{T}\mathbf{x} - \hat{\mathfrak{m}}_j^{(\text{spca})}\|^2. \tag{1}$$

As a consequence, the discriminating power of both PCA and SPCA directly relates to the limiting (squared) distances $\Delta\mathfrak{m}_{(j,j')}^{(\times)} \equiv \|\mathfrak{m}_j^{(\times)} - \mathfrak{m}_{j'}^{(\times)}\|^2$, for all pairs of class indices $1 \leq j \neq j' \leq m$, and the classification error $P(\mathbf{x} \to \mathcal{C}_{j'} | \mathbf{x} \in \mathcal{C}_j)$ satisfies

$$P(\mathbf{x} \to \mathcal{C}_{j'} | \mathbf{x} \in \mathcal{C}_j) = \mathcal{Q}\left(\frac{1}{2}\sqrt{\Delta\mathfrak{m}_{(j,j')}^{(\times)}}\right) + o(1), \quad \text{for} \quad \mathcal{Q}(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-x^2} dx.$$

In particular, and as confirmed by Figure 1, when $c_j = c_{j'}$, SPCA uniformly dominates PCA:

$$\Delta\mathfrak{m}_{(j,j')}^{(\text{spca})} - \Delta\mathfrak{m}_{(j,j')}^{(\text{pca})} = \sum_{i=1}^{\tau} \frac{\left(\bar{v}_i^\mathsf{T} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} (e_j^{[\tau]} - e_{j'}^{[\tau]})\right)^2}{\ell_i^2(\ell_i + 1)} \geq 0.$$

For $m = 2$ classes, irrespective of c_1, c_2, one even finds in explicit form

$$\Delta\mathfrak{m}_{(1,2)}^{(\text{spca})} - \Delta\mathfrak{m}_{(1,2)}^{(\text{pca})} = \frac{16}{\frac{n}{p}\|\Delta\mu\|^2 + 4}, \qquad \frac{\Delta\mathfrak{m}_{(1,2)}^{(\text{spca})} - \Delta\mathfrak{m}_{(1,2)}^{(\text{pca})}}{\Delta\mathfrak{m}_{(1,2)}^{(\text{spca})}} = \frac{16}{\frac{n}{p}\|\Delta\mu\|^4}$$

where $\Delta\mu \equiv \mu_1 - \mu_2$, conveniently showing the influence of n/p and of $\|\Delta\mu\|^2$ in the relative performance gap, which vanishes as the task gets easier or as n/p increases (so with more data).

Summarizing, under a large dimensional setting, we showed that SPCA-based classification uniformly outperform the PCA alternative, thus motivating the design of an SPCA-based MTL approach.

4 From single- to multi-task SPCA-based learning

4.1 Multi-class setting

Let now $X = [X_{[1]}, \ldots, X_{[k]}] \in \mathbb{R}^{p \times n}$ be a collection of n independent p-dimensional data vectors, divided into k subsets attached to individual "tasks". Task t is an m-class classification problem with training samples $X_{[t]} = [X_{[t]1}, \ldots, X_{[t]m}] \in \mathbb{R}^{p \times n_i}$ with $X_{[t]j} = [x_{t1}^{(j)}, \ldots, x_{tn_{tj}}^{(j)}] \in \mathbb{R}^{p \times n_{tj}}$ the n_{tj} vectors of class $j \in \{1, \ldots, m\}$. In particular, $n = \sum_{t=1}^k n_t$ for $n_t \equiv \sum_{j=1}^m n_{tj}$.

To each $x_{t\ell}^{(j)} \in \mathbb{R}^p$ is attached a corresponding "label" (or score) $y_{t\ell}^{(j)} \in \mathbb{R}^m$. We denote in short $y_t = [y_{t1}^{(1)}, \ldots, y_{tn_t}^{(m)}]^\mathsf{T} \in \mathbb{R}^{n_t \times m}$ and $y = [y_1^\mathsf{T}, \ldots, y_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{n \times m}$ the matrix of all labels. The natural MTL extension of SPCA would default $y_{t\ell}^{(j)} \in \mathbb{R}^m$ to the canonical vectors $e_j^{[m]}$ (or to ± 1 in the binary case). We disrupt here from this approach by explicitly *not* imposing a value for $y_{t\ell}^{(j)}$: this will be seen to be key to *avert the problem of negative transfer*. We only let $y_{t\ell}^{(j)} = \tilde{y}_{tj}$, for all $1 \leq \ell \leq n_{tj}$ and for some generic matrix $\tilde{y} = [\tilde{y}_{11}, \ldots, \tilde{y}_{km}]^\mathsf{T} \in \mathbb{R}^{mk \times m}$, i.e., we impose that

$$y = J\tilde{y}, \quad \text{for} \quad J = [j_{11}, \ldots, j_{mk}], \quad \text{where} \quad j_{tj} = (0, \ldots, 0, \mathbb{1}_{n_{tj}}, 0, \ldots, 0)^\mathsf{T}.$$

As with the single-task case, we work under a Gaussian mixture model for each class $\mathcal{C}_{tj}$.

Assumption 3 (Distribution of X) *For class j of Task t, denoted $\mathcal{C}_{tj}$, $x_{t\ell}^{(j)} \sim \mathcal{N}(\mu_{tj}, I_p)$, for some $\mu_{tj} \in \mathbb{R}^p$. We further denote $M \equiv [\mu_{11}, \ldots, \mu_{km}] \in \mathbb{R}^{p \times mk}$.*

Assumption 4 (Growth Rate) *As $n \to \infty$, $p/n \to c_0 > 0$ and, for $1 \leq j \leq m$, $n_{tj}/n \to c_{tj} > 0$. Denoting $c = [c_{11}, \ldots, c_{km}]^\mathsf{T} \in \mathbb{R}^{km}$ and $\mathcal{D}_c = \mathrm{diag}(c)$, $(1/c_0)\mathcal{D}_c^{\frac{1}{2}} M^\mathsf{T} M \mathcal{D}_c^{\frac{1}{2}} \to \mathcal{M} \in \mathbb{R}^{mk \times mk}$.*

We are now in position to present the main technical result of the article.

Theorem 3 (MTL Supervised Principal Component Analysis) *Let $\mathbf{x} \sim \mathcal{N}(\mu_{tj}, I_p)$ independent of X and $V \in \mathbb{R}^{p \times \tau}$ be the collection of the $\tau \leq mk$ dominant eigenvectors of $\frac{Xyy^\mathsf{T}X^\mathsf{T}}{np} \in \mathbb{R}^{p \times p}$. Then, under Assumptions 3-4, as $p, n \to \infty$, in probability,*

$$V^\mathsf{T}\mathbf{x} - G_{tj} \to 0, \quad G_{tj} \sim \mathcal{N}(\mathfrak{m}_{tj}, I_\tau), \quad \text{for} \quad [\mathfrak{m}_{tj}]_i = \sqrt{1/(c_0\tilde{\ell}_i)}\, \bar{v}_i^\mathsf{T} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} e_{tj}^{[mk]}$$

with $\tilde{\ell}_1 > \ldots > \tilde{\ell}_{mk}$ the eigenvalues of $(\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}}(\mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{\frac{1}{2}} + \mathcal{D}_c)(\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}}$ and $\bar{v}_1, \ldots, \bar{v}_{mk}$ their eigenvectors.[4]

As in the single task case, despite the high dimension of the data statistics appearing in V, the asymptotic performance only depends on the (small) $mk \times mk$ matrices $\mathcal{M}$ and $\mathcal{D}_c$, which here leverages the inter-task inter-class products $\mu_{tj}^\mathsf{T}\mu_{t'j'}$. This correlation between tasks *together with the labelling choice $\tilde{y}$* (importantly recall that here $V = V(y)$) influences the MTL performance. The next section discusses how to optimally *align $\tilde{y}$ and $\mathcal{M}$* so to maximize this performance. This, in addition to Remark 1 being evidently still valid here (i.e., c and $\mathcal{M}$ can be a priori consistently estimated), will unfold into our proposed asymptotically optimal MTL SPCA algorithm.

4.2 Binary classification and optimal labels

To obtain more telling conclusions, let us now focus on binary classification ($m = 2$). In this case, $y = J\tilde{y}$, with $\tilde{y} \in \mathbb{R}^{2k}$ (rather than in $\mathbb{R}^{2k \times 2}$) unidimensional. Here $\frac{Xyy^\mathsf{T}X^\mathsf{T}}{np}$ has for unique non-trivial eigenvector $Xy/\|Xy\|$ and $V^\mathsf{T}\mathbf{x}$ is scalar.

Corollary 1 (Binary MTL Supervised Principal Component Analysis) *Let $\mathbf{x} \sim \mathcal{N}(\mu_{tj}, I_p)$ independent of X. Then, under Assumptions 3-4 and the above setting, as $p, n \to \infty$,*

$$V^\mathsf{T}\mathbf{x} - G_{tj} \to 0, \quad G_{tj} \sim \mathcal{N}(\mathfrak{m}_{tj}^{(\mathrm{bin})}, 1), \quad \text{where} \quad \mathfrak{m}_{tj}^{(\mathrm{bin})} = \frac{\tilde{y}^\mathsf{T}\mathcal{D}_c^{\frac{1}{2}}\mathcal{M}\mathcal{D}_c^{-\frac{1}{2}} e_{tj}}{\sqrt{\tilde{y}^\mathsf{T}(\mathcal{D}_c^{\frac{1}{2}}\mathcal{M}\mathcal{D}_c^{\frac{1}{2}} + \mathcal{D}_c)\tilde{y}}}.$$

From Corollary 1, denoting $\hat{\mathfrak{m}}_{t1}^{(\mathrm{bin})}$ the natural consistent estimate for $\mathfrak{m}_{t1}^{(\mathrm{bin})}$ (as per Remark 1), the optimal class allocation decision for $\mathbf{x}$ reduces to the "averaged-mean" test

$$V^\mathsf{T}\mathbf{x} = V(y)^\mathsf{T}\mathbf{x} \underset{\mathcal{C}_{t2}}{\overset{\mathcal{C}_{t1}}{\gtrless}} \frac{1}{2}\left(\hat{\mathfrak{m}}_{t1}^{(\mathrm{bin})} + \hat{\mathfrak{m}}_{t2}^{(\mathrm{bin})}\right) \tag{2}$$

[4]For simplicitly, we avoid the scenario where the eigenvalues $\tilde{\ell}_j$ appear with multiplicity, which would require to gather the eigenvectors into eigenspaces. This would in effect only make the notations more cumbersome.

with corresponding classification error rate $\epsilon_t \equiv \frac{1}{2}P(\mathbf{x} \to \mathcal{C}_{t2}|\mathbf{x} \in \mathcal{C}_{t1}) + \frac{1}{2}P(\mathbf{x} \to \mathcal{C}_{t1}|\mathbf{x} \in \mathcal{C}_{t2})$ (assuming equal prior class probability) given by

$$\epsilon_t \equiv P\left(V^\mathsf{T}\mathbf{x} \underset{\mathcal{C}_{t2}}{\overset{\mathcal{C}_{t1}}{\gtrless}} \frac{1}{2}(\hat{\mathfrak{m}}_{t1}^{(\mathrm{bin})} + \hat{\mathfrak{m}}_{t2}^{(\mathrm{bin})})\right) = \mathcal{Q}\left(\frac{1}{2}(\mathfrak{m}_{t1}^{(\mathrm{bin})} - \mathfrak{m}_{t2}^{(\mathrm{bin})})\right) + o(1). \tag{3}$$

From the expression of $\mathfrak{m}_{tj}^{(\mathrm{bin})}$, the asymptotic performance clearly depends on a proper choice of $\tilde{y}$. This expression being quadratic in $\tilde{y}$, the ϵ_t minimizer $\tilde{y} = \tilde{y}_{[t]}^\star$ assumes a closed-form:

$$\tilde{y}_{[t]}^\star \equiv \underset{\tilde{y}\in\mathbb{R}^{2k}}{\arg\max}\, (\mathfrak{m}_{t1}^{(\mathrm{bin})} - \mathfrak{m}_{t2}^{(\mathrm{bin})})^2 = \mathcal{D}_c^{-\frac{1}{2}} \left(\mathcal{M} + I_{2k}\right)^{-1} \mathcal{M}\mathcal{D}_c^{-\frac{1}{2}}(e_{t1} - e_{t2}).$$

Letting $\hat{\tilde{y}}_{[t]}^\star$ be the natural consistent estimator of $\tilde{y}_{[t]}^\star$ (again from Remark 1), and updating $V = V(\tilde{y}_{[t]})$ accordingly, the corresponding (asymptotically) optimal value $\epsilon_t^\star$ of the error rate ϵ_t is

$$\epsilon_t^\star = \mathcal{Q}\left(\frac{1}{2}\sqrt{(e_{t1}^{[2k]} - e_{t2}^{[2k]})^\mathsf{T}\mathcal{D}_c^{-\frac{1}{2}}\mathcal{M}\left(\mathcal{M} + I_{2k}\right)^{-1}\mathcal{M}\mathcal{D}_c^{-\frac{1}{2}}(e_{t1}^{[2k]} - e_{t2}^{[2k]})}\right) + o(1). \tag{4}$$

This formula is instructive to discuss: under strong or weak task correlation, $\tilde{y}_{[t]}^\star$ implements differing strategies to avoid *negative transfers*. For instance, if $\mu_{tj}^\mathsf{T}\mu_{t'j'} = 0$ for all $t' \neq t$ and $j, j' \in \{1, \ldots, m\}$, then the two rows and columns of $\mathcal{M}$ associated to Task t are all zero but on the 2×2 diagonal block: $\tilde{y}_{[t]}^\star$ is then all zeros but on its two Task-t elements; any other value at these zero-entry locations (such as the usual ± 1) is suboptimal and possibly severely detrimental to classification. Letting $\tilde{y}_{[t]} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ is even more detrimental when $\mu_{tj}^\mathsf{T}\mu_{t'j'} < 0$ for some $t' \neq t'$: when the mapping of classes across tasks is reversed, these tasks work *against* the classification.

Remark 2 (On Bayes optimality) *Under the present MTL setting of a mixture of two isotropic random Gaussian vectors, the authors recently established that the* Bayes optimal *error rate (associated to the decision rule* $\inf_g P(g(\mathbf{x}) > 0 \mid \mathbf{x} \in \mathcal{C}_{t1})$*) precisely* coincides *with* $\varepsilon_{t1}^\star$.[5] *This proves here that, at least under the present data configuration, the proposed SPCA-MTL framework is optimal.*

4.3 Binary-based multi-class classification

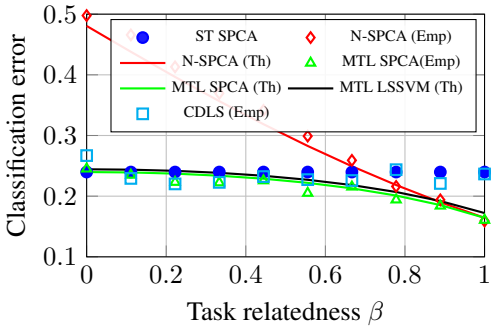
Having an optimal binary classification framework for every task and every pair of classes, one may expect to reach high performance levels in generic multi-class settings by resorting to a *one-versus-all* extension of the binary case. For every target task t, one-versus-all implements m binary classifiers: classifier $\ell \in \{1, \ldots, m\}$ separates class $\mathcal{C}_{t\ell}$ – locally renamed "class $\mathcal{C}_{t1}^{(\ell)}$" – from all other classes – gathered as a unique "class $\mathcal{C}_{t2}^{(\ell)}$". Each binary classifier is then "optimized" using labels $\tilde{y}_{[t]}^{\star(\ell)}$ as per Equation (4); however, the joint class $\mathcal{C}_{t2}^{(\ell)}$ is here composed of a Gaussian *mixture*: this disrupts with our optimal framework, thereby in general leading to suboptimal labels; in practice though, for sufficiently distinct classes, the (suboptimal) label $\tilde{y}_{[t]}^{\star(\ell)}$ manages to isolate the value $\mathfrak{m}_{t\ell}^{(\mathrm{bin})} = \mathfrak{m}_{t1}^{(\mathrm{bin},\ell)}$ for class $\mathcal{C}_{t\ell} = \mathcal{C}_{t1}^{(\ell)}$ from the values $\mathfrak{m}_{tj}^{(\mathrm{bin})}$ of all other classes $\mathcal{C}_{tj}$, $j \neq \ell$, to such an extent that (relatively speaking) these $\mathfrak{m}_{tj}^{(\mathrm{bin})}$ can be considered quite close, and so close to their mean $\mathfrak{m}_{t2}^{(\mathrm{bin},\ell)}$, without much impact on the classifier performance. Finally, the class allocation for unknown data $\mathbf{x}$ is based on a largest classifier-score. But, to avoid biases which naturally arise in the one-versus-all approach [9, Section 7.1.3], this imposes that the m different classifiers be "comparable and aligned". To this end, we exploit Corollary 1 and Remark 1 which give a consistent estimate of all classifier statistics: the test scores for each classifier can be centered so that the asymptotic distribution for class $\mathcal{C}_{t1}^{(\ell)}$ is a *standard normal distribution for each* $1 \leq \ell \leq m$, thereby automatically discarding biases. Thus, instead of selecting the class with largest score $\arg\max_\ell V(y_{[t]}^{\star(\ell)})^\mathsf{T}\mathbf{x}$ (as conventionally performed [9, Section 7.1.3]), the class allocation is based on the centered scores $\arg\max_\ell\{V(y_{[t]}^{\star(\ell)})^\mathsf{T}\mathbf{x} - \mathfrak{m}_{t1}^{(\mathrm{bin},\ell)}\}$.[6] These discussions result in Algorithm 1.

[5] The result builds on recent advances in physics-inspired (spin glass models) large dimensional statistics; see for instance [26] for a similar result in a single task semi-supervised learning setting. Being a parallel work of the same authors, the reference is concealed in the present version to maintain anonymity.

[6] More detail and illustrations are provided in the supplementary material.

Algorithm 1: Proposed multi-class MTL SPCA algorithm.

Input: Training $X = [X_{[1]}, \ldots, X_{[k]}]$, $X_{[t']} = [X_{[t']1}, \ldots, X_{[t']m}]$, $X_{[t']\ell} \in \mathbb{R}^{p \times n_{t'\ell}}$ and test $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for target Task t.
Center and normalize the data per task using z-score normalization [36].
for $\ell = 1$ **to** m **do**
  Estimate c and $\mathcal{M}$ (from Remark 1) using $X_{[t']\ell}$ as data of class $\mathcal{C}_{t'1}^{(\ell)}$ for each $t' \in \{1, \ldots, k\}$ and $\{X_{[t']1}, \ldots, X_{[t']m}\} \setminus \{X_{[t']\ell}\}$ as data of class $\mathcal{C}_{t'2}^{(\ell)}$.
  Evaluate labels $\tilde{y}_{[t]}^{\star(\ell)} = \mathcal{D}_c^{-\frac{1}{2}} (\mathcal{M} + I_{2k})^{-1} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} (e_{t1}^{[2k]} - e_{t2}^{[2k]})$.
  Compute the classification score $g_{\mathbf{x},t}^{(\ell)} = \tilde{y}_{[t]}^{\star(\ell)\mathsf{T}} J^{\mathsf{T}} X^{\mathsf{T}} \mathbf{x} / \| \tilde{y}_{[t]}^{\star(\ell)\mathsf{T}} J^{\mathsf{T}} X^{\mathsf{T}} \|$.
  Estimate $\mathfrak{m}_{t1}^{(\mathrm{bin},\ell)}$ as $\hat{\mathfrak{m}}_{t1}^{(\mathrm{bin},\ell)}$ from Corollary 1.
end for
Output: $\hat{\ell} = \arg\max_{\ell \in \{1,\ldots,m\}} (g_{\boldsymbol{x},t}^{(\ell)} - \hat{\mathfrak{m}}_{t1}^{(\mathrm{bin},\ell)})$.



p	MTL SPCA	MTL LSSVM	CDLS
16	0.34 s	4.15 s	7.16 s
32	0.34 s	4.46 s	7.43 s
64	0.39 s	5.38 s	8.61 s
128	0.40 s	8.28 s	8.80 s
256	0.55 s	12.2 s	11.9 s
512	0.57 s	48.3 s	17.5 s
1024	0.88 s	315.6 s	27.1 s
2048	2.02 s	1591.8 s	73.5 s

Figure 2: **(Left)** Theoretical (Th)/empirical (Emp) error rate for 2-class Gaussian mixture transfer with means $\mu_1 = e_1^{[p]}$, $\mu_1^{\perp} = e_p^{[p]}$, $\mu_2 = \beta\mu_1 + \sqrt{1-\beta^2}\mu_1^{\perp}$, $p = 100$, $n_{1j} = 1\,000$, $n_{2j} = 50$; **(Right)** running time comparison (in sec); $n = 2p$, $n_{tj}/n = 0.25$. Averaged over 1 000 test samples.

5 Supporting experiments

We here compare the performance of Algorithm 1 (MTL SPCA), on both synthetic and real data benchmarks, to competing state-of-the-art methods, such as MTL-LSSVM [46] and CDLS [21].[7]

Transfer learning for binary classification. First consider a two-task two-class ($k, m = 2$) scenario with $x_{t\ell}^{(j)} \sim \mathcal{N}((-1)^j \mu_t, I_p)$, $\mu_2 = \beta\mu_1 + \sqrt{1-\beta^2}\mu_1^{\perp}$ for $\mu_1^{\perp}$ any vector orthogonal to μ_1 and $\beta \in [0, 1]$ controlling inter-task similarity. Figure 2 depicts the empirical and theoretical classification error ϵ_2 for the above methods for $p = 100$ and $n = 2\,200$; for completeness, the single-task SPCA (ST-SPCA) of Section 3 (which disregards data from other tasks) as well as its naive MTL extension with labels $\tilde{y}_{[t]} = [1, -1, \ldots, 1, -1]^{\mathsf{T}}$ (N-SPCA) were added. MTL SPCA properly tracks task relatedness, while CDLS fails when both tasks are quite similar. MTL LSSVM shows identical performances but at the cost of setting optimal hyperparameters. Probably most importantly, when *not optimizing* the labels y, the performance (of N-SPCA) is strongly degraded by *negative transfer*, particularly when tasks are not related. Figure 2 also provides typical computational times for each algorithm when run on a modern laptop, and confirms that Algorithm 1 scales very favorably with the data dimension p, while MTL LSSVM and CDLS quickly become prohibitively expensive.

Transfer learning for multi-class classification. We next experiment on the ImageClef dataset [22] made of 12 common categories shared by 3 public data "domains": Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). Every pair of domains is successively selected as

[7] We insist that MTL SPCA is intended to function under the constraint of scarce data and does not account for the very nature of these data: to avoid arbitrary conclusions, image- or language-dedicated MTL and transfer learning methods (e.g., modern adaptions of deep nets for transfer learning [45]) are not used for comparison.

S/T	P→I	P→C	I→P	I→C	C→P	C→I	Average
ST SPCA	91.84	96.24	82.26	96.24	82.26	91.84	90.11
N-SPCA	92.21	96.37	84.34	95.97	81.34	90.47	90.12
MTL LSSVM	*93.03*	**97.24**	84.79	**97.74**	*83.74*	**94.92**	**91.91**
CDLS	92.03	94.62	*84.82*	95.72	81.04	92.54	90.13
MTL SPCA	**93.39**	*96.61*	**85.24**	*96.68*	**83.76**	*93.39*	*91.51*

Table 1: Transfer learning accuracy for the ImageClef database: P(Pascal), I(Imagenet), C(Caltech); different "Source to target" task pairs (S→T) based on Resnet-50 features.

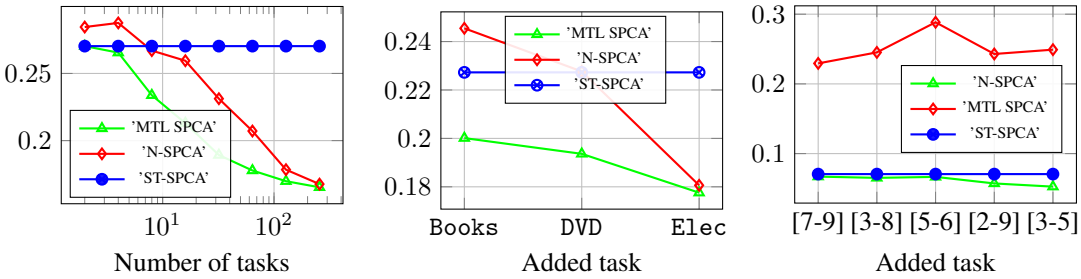


Figure 3: Empirical classification error vs. number of tasks; **(Left)** Synthetic Gaussian with random task correlation: $p = 200$, $n_{11} = n_{12} = 50$, $n_{21} = n_{22} = 5$, 10 000 test samples; **(Center)** Amazon Review: $n_{11} = n_{12} = 100, n_{21} = n_{22} = 50$, 2 000 test samples; **(Right)** MNIST: initial $p = 100$-PCA preprocessing, $n_{11} = n_{12} = 100$, $n_{21} = n_{22} = 50$, 500 test samples.

"source" and a "target" for binary (transfer) multi-task learning, resulting in 6 transfer tasks S→T for S,T∈ {I,C,P}. Table 1 supports the stable and competitive performance of MTL-SPCA, on par with MTL LSSVM (but much cheaper).

Increasing the number of tasks. We now investigate the comparative gains induced when increasing the number of tasks. To best observe the reaction of each algorithm to the additional tasks, we here consider both a tunable synthetic Gaussian mixture and (less tractable) real-world data. The synthetic data consist of two Gaussian classes with means $\mu_{tj} = (-1)^j \mu_{[t]}$ with $\mu_{[t]} = \beta_{[t]}\mu + \sqrt{1-\beta_{[t]}^2}\mu^\perp$ for $\beta_{[t]}$ drawn uniformly at random in $[0,1]$ and with $\mu = e_1^{[p]}$, $\mu^\perp = e_p^{[p]}$. The real-world data are the Amazon review (textual) dataset[8] [10] and the MNIST (image) dataset [14]. For Amazon review, the positive vs. negative reviews of "`books`", "`dvd`" and "`electronics`" products are added to help classify the positive vs. negative reviews of "`kitchen`" products. For MNIST, additional digit pairs are added progressively to help classify the target pair $(1, 4)$. The results are shown in Figure 3 which confirms that (i) the naive extension of SPCA (N-SPCA) with labels ± 1 can fail to the point of being bested by (single task) ST-SPCA, (ii) MTL-SPCA never decays with more tasks.

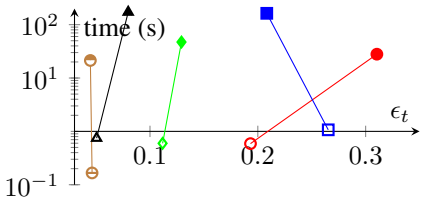
Multi-class multi-task classification. We finally turn to the full multi-task multi-class setting of Algorithm 1. Figure 4 simultaneously compares running time and error rates of MTL-SPCA and MTL-LSSVM[9] on a variety of multi-task datasets, and again confirms the overall computational gains (by decades!) of MTL-SPCA for approximately the same performance levels.

6 Conclusion

Following recent works on large dimensional statistics for the design of simple, cost-efficient, and tractable machine learning algorithms [13], the article confirms the possibility to achieve high performance levels while theoretically averting the main sources of biases, here for the a priori difficult concept of multi-task learning. The article, we hope, will be followed by further investigations of sustainable AI algorithms, driven by modern mathematical tools. In the present multi-task learning

[8] Encoded in $p = 400$-dimensional tf*idf feature vectors of bag-of-words unigrams and bigrams.

[9] CDLS only handles multi-task learning with $k = 2$ and cannot be used for comparison.



Datasets (Features)	Tasks	Classes	Mark
Synthetic (Gaussian)	3	10	○
Office-Caltech[17] (VGG)	4	10	◇
Office 31[42] (Resnet-50)	4	31	□
Office-Home[47] (Resnet-50)	3	65	△
Image-Clef[22] (Resnet-50)	3	12	⊖

Figure 4: **(Left)** Runtime vs. classification error (ϵ_t) for multi-task multi-class MTL-LSSVM (filled marks) and MTL-SPCA (empty marks). **(Right)** Datasets. Synthetic: $\mu_j = 2e_j^{[p]}$, $\mu_j^{\perp} = 2e_{p-j}^{[p]}$, $\beta_1 = 0.2$, $\beta_2 = 0.4$, $\beta_3 = 0.6$; $p = 200$, $n_{1j} = n_{2j} = 100$, $n_{3j} = 50$; 1 000 test sample averaging.

framework, practically realistic extensions to semi-supervised learning (when labelled data are scarce) with possibly missing, unbalanced, or incorrectly labelled data are being considered by the authors.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2020/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to autmoatically hide this section in the anonymized submission.

References

[1] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

[3] Hassan Ashtiani and Ali Ghodsi. A dimension-independent generalization bound for kernel supervised principal component analysis. In *Feature Extraction: Modern Questions and Challenges*, pages 19–29. PMLR, 2015.

[4] Jinho Baik and Jack W Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of multivariate analysis*, 97(6):1382–1408, 2006.

[5] Eric Bair, Trevor Hastie, Debashis Paul, and Robert Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137, 2006.

[6] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013.

[7] Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

[8] Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

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

[10] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

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

[12] Guoqing Chao, Yuan Luo, and Weiping Ding. Recent advances in supervised dimension reduction: A survey. *Machine learning and knowledge extraction*, 1(1):341–358, 2019.

[13] Romain Couillet, Florent Chatelain, and Nicolas Le Bihan. Two-way kernel matrix puncturing: towards resource-efficient pca and spectral clustering. *arXiv preprint arXiv:2102.12293*, 2021.

[14] 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.

[15] 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.

[16] Benyamin Ghojogh and Mark Crowley. Unsupervised and supervised principal component analysis: Tutorial. *arXiv preprint arXiv:1906.03148*, 2019.

[17] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

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

[19] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[20] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

[21] 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.

[22] Bogdan Ionescu, Henning Müller, Mauricio Villegas, Helbert Arenas, Giulia Boato, Duc-Tien Dang-Nguyen, Yashin Dicente Cid, Carsten Eickhoff, Alba G Seco de Herrera, Cathal Gurrin, et al. Overview of imageclef 2017: Information extraction from images. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 315–337. Springer, 2017.

[23] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.

[24] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[25] Seunggeun Lee, Fei Zou, and Fred A Wright. Convergence and prediction of principal component scores in high-dimensional settings. *Annals of statistics*, 38(6):3605, 2010.

[26] 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.

[27] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2, 1-norm minimization. *arXiv preprint arXiv:1205.2631*, 2012.

[28] Qiuhua Liu, Xuejun Liao, and Lawrence Carin. Semi-supervised multitask learning. *Advances in Neural Information Processing Systems*, 20:937–944, 2007.

[29] Mingsheng Long, Jianmin Wang, Guiguang Ding, Dou Shen, and Qiang Yang. Transfer learning with graph co-regularization. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1805–1818, 2013.

[30] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

[31] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

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

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

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

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

[36] S Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[37] Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, pages 1617–1642, 2007.

[38] Marek Rei. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*, 2017.

[39] Alexander Ritchie, Clayton Scott, Laura Balzano, Daniel Kessler, and Chandra S Sripada. Supervised principal component analysis via manifold optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 6–10. IEEE, 2019.

[40] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

[41] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[42] 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.

[43] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[44] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[45] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[46] Malik Tiomoko, Romain Couillet, and Hafiz Tiomoko. Large dimensional analysis and improvement of multi task learning. *arXiv preprint arXiv:2009.01591*, 2020.

[47] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

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

[49] 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.

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

[51] Xinyi Zhang, Qiang Sun, and Dehan Kong. Supervised principal component regression for functional response with high dimensional predictors. *arXiv preprint arXiv:2103.11567*, 2021.

[52] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

[53] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[54] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

[55] Yu Zhang and Dit-Yan Yeung. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):1–31, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not informative and disturb the readability of the graphs/tables.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material

Malik Tiomoko
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
91190, Gif-sur-Yvette, France. malik.tiomoko@u-psud.fr

Romain Couillet
Gipsa Lab
Université Grenoble Alpes
romain.couillet@gipsa-lab.grenoble-inp.fr

Frédéric Pascal
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes
91190, Gif-sur-Yvette, France
frederic.pascal@centralesupelec.fr

Abstract

This document contains the main technical arguments omitted in the core of the article due to space limitation and is organized as follows. Section 1 details the large dimensional analysis of PCA. Section 2 provides the asymptotic performance of SPCA in the most general case of a Gaussian mixture model (with arbitrary means and covariances) in a multi-task setting. The single-task setting is retrieved as a special case. Section 3 details and illustrates the binary-based multi-class classification and proposes alternative schemes to the one-versus-all approach covered in the main article. Supplementary experiments are provided in Section 4.

1 Large dimensional analysis of Single Task PCA

We recall that the solution U of PCA is explicitly given by the collection of the eigenvectors associated with the τ largest eigenvalues of $\frac{1}{p}XX^\mathsf{T}$. The goal of this section is to compute the isolated eigenvalues of $\frac{1}{p}XX^\mathsf{T}$ and to study the behavior of the projection of a new test data on the feature space spanned by PCA under the large dimensional regime.

Assumption 1 (Distribution of X and $\mathbf{x}$) *The columns of X are independent random vectors with $X = [X_1, \ldots, X_m]$, $X_j = [x_1^{(j)}, \ldots, x_{n_j}^{(j)}] \in \mathbb{R}^{p \times n_j}$ where $x_i^{(j)} \sim \mathcal{N}(\mu_j, I_p)$. As for $\mathbf{x}$, it follows an independent $\mathcal{N}(\mu_\mathbf{x}, I_p)$ distribution. We will further denote $x \in \mathcal{C}_j$ to indicate that data vector x belongs to class j, i.e., $x \sim \mathcal{N}(\mu_j, I_p)$.*

Assumption 2 (Growth Rate) *As $n \to \infty$, $p/n \to c_0 > 0$ and, for $1 \leq j \leq m$, $\frac{n_j}{n} \to c_j > 0$; we will denote $c = [c_1, \ldots, c_m]^\mathsf{T}$. Furthermore, the latent feature space dimension τ is constant with respect to n, p.*

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

1.1 Isolated eigenvalues

To retrieve the isolated eigenvalues of $\frac{1}{p}XX^\mathsf{T}$, we simply aim to solve the determinant equation in $z \in \mathbb{R}_+$

$$\det\left(\frac{1}{p}XX^\mathsf{T} - zI_p\right) = 0.$$

Writing $X = MJ^\mathsf{T} + W$ with $M = [\mu_1, \ldots, \mu_m] \in \mathbb{R}^{p\times m}$, $J = [j_1, \ldots, j_m]$, where $j_j = (0, \ldots, 0, \mathbb{1}_{n_j}, 0, \ldots, 0)^\mathsf{T}$ and where W is a random matrix with independent standard Gaussian entries, this becomes

$$\det\left(\frac{1}{p}WW^\mathsf{T} + \mathcal{U}\mathcal{V}^\mathsf{T} - zI_p\right) = 0, \tag{1}$$

where $\mathcal{U} = \frac{1}{\sqrt{p}}[M, WJ] \in \mathbb{R}^{p\times 2m}$ and $\mathcal{V} = \frac{1}{\sqrt{p}}[MJ^\mathsf{T}J + W^\mathsf{T}J, M] \in \mathbb{R}^{p\times 2m}$ are low rank matrices (as $n, p \to \infty$); as for $\frac{1}{p}WW^\mathsf{T}$, its limiting eigenvalue distribution under Assumption 2 is known as the Marčenko-Pastur law [5], recalled next in whole generality:

Theorem 1 *Let W be a $p \times n$ matrix with i.i.d. real- or complex-valued entries with zero mean and unit variance. Then, as $n, p \to \infty$ such that $p/n \xrightarrow{\text{a.s.}} c_0$, the* empirical spectral measure *$\mu_{\hat{C}} = \frac{1}{p}\sum_{i=1}^p \delta_{\hat{\lambda}_i}$ of the eigenvalues $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_p$ of $\frac{1}{p}WW^\mathsf{T}$, converges weakly, with probability one, to a nonrandom distribution, known as the Marčenko–Pastur law and denoted $\mu_{\mathrm{MP}}^{c_0}$. If $c_0 \in (0, 1)$, $\mu_{\mathrm{MP}}^{c_0}$ has density:*

$$\mu_{\mathrm{MP}}^{c_0}(dx) = \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{2\pi c_0 x}dx$$

where $\lambda_\pm = (1 \pm \sqrt{1/c_0})^2$. If $c_0 \in (1, \infty)$, μ_{MP} is the weighted sum of a point mass at 0 and of the density $\mu_{\mathrm{MP}}^{1/c_0}$ with weights $1 - (1/c_0)$ and $1/c_0$.

The spectrum of $\frac{1}{p}WW^\mathsf{T}$, which contains no structural information (generally refer as a "noise bulk"), will not be of interest for classification. The challenge is to determine which observed eigenvalues actually represent the class structure. Specifically, let us seek for the presence of an eigenvalue λ_j of $\frac{1}{p}XX^\mathsf{T}$ asymptotically greater than the limit $(1 + \sqrt{1/c_0})^2$ of the largest eigenvalue of $\frac{1}{p}WW^\mathsf{T}$. Following the initial ideas of [1, 2], the approach is to isolate the low rank contribution $\mathcal{U}\mathcal{V}^\mathsf{T}$ from the noise matrix $\frac{1}{p}WW^\mathsf{T}$. Factoring out $\frac{1}{p}WW^\mathsf{T} - zI_p$ and using Sylverster's identity ($\det(AB + I) = \det(BA + I)$), Equation (1) is equivalent to:

$$\det\left(\mathcal{V}^\mathsf{T}Q(z)\mathcal{U} + I_{2m}\right) = 0, \quad \text{with} \quad Q(z) = \left(\frac{1}{p}WW^\mathsf{T} - zI_p\right)^{-1}.$$

We next retrieve the large dimensional limit (or, more specifically a *deterministic equivalent* [4, Chapter 6]) of $\mathcal{V}^\mathsf{T}Q(z)\mathcal{U} + I_{2m}$ under Assumptions 1 and 2. Defining the *resolvents* and *co-resolvents* $Q(z) = (\frac{1}{p}WW^\mathsf{T} - zI_p)^{-1}$ and $\tilde{Q}(z) = (\frac{1}{p}W^\mathsf{T}W - zI_n)^{-1}$, as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$, we have

$$Q(z) \leftrightarrow \bar{Q}(z),\ \bar{Q}(z) = \delta(z)I_p$$
$$\tilde{Q}(z) \leftrightarrow \bar{\tilde{Q}}(z),\ \bar{\tilde{Q}}(z) = \tilde{\delta}(z)I_n$$

where $(\tilde{\delta}(z), \delta(z))$ are defined as

$$\delta(z) = \frac{c_0 - 1 - c_0 z + \sqrt{(c_0 - 1 - c_0 z)^2 - 4z}}{2z}, \qquad \tilde{\delta}(z) = \frac{1}{c_0}\left(\delta(z) + \frac{1 - c_0}{z}\right)$$

and the notation $F \leftrightarrow \bar{F}$ stands for the fact that, under Assumption 2, for any deterministic linear functional $f : \mathbb{R}^{n\times p} \to \mathbb{R}$, $f(F - \bar{F}) \to 0$ almost surely (for instance, 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, $\frac{1}{n}\mathrm{tr}A(F - \bar{F}) \xrightarrow{\text{a.s.}} 0$).

In particular, developing the definitions of $\mathcal{V}$ and $\mathcal{U}$,

$$\det\left(\mathcal{V}^\mathsf{T}Q(z)\mathcal{U}+I_{2m}\right)$$
$$=\det\begin{pmatrix} I_m+\frac{1}{p}J^\mathsf{T}JM^\mathsf{T}Q(z)M+\frac{1}{p}J^\mathsf{T}W^\mathsf{T}Q(z)M & \frac{1}{p}J^\mathsf{T}JM^\mathsf{T}Q(z)WJ+\frac{1}{p}J^\mathsf{T}W^\mathsf{T}Q(z)WJ \\ \frac{1}{p}M^\mathsf{T}Q(z)M & I_m+\frac{1}{p}M^\mathsf{T}Q(z)WJ \end{pmatrix}$$

and we then have, from the above deterministic equivalents, that

$$\begin{aligned}\det\left(\mathcal{V}^\mathsf{T}Q(z)\mathcal{U}+I_{2m}\right) &= \det\begin{pmatrix} I_m+\delta(z)\frac{J^\mathsf{T}J}{p}M^\mathsf{T}M & (1+z\tilde{\delta}(z))J^\mathsf{T}J \\ \delta(z)\frac{1}{p}M^\mathsf{T}M & I_m \end{pmatrix}+o(1)\\ &= \det\left(I_m-z\tilde{\delta}(z)\delta(z)\frac{J^\mathsf{T}J}{p}M^\mathsf{T}M\right)+o(1).\end{aligned}$$

The limiting position of the (hypothetical) isolated eigenvalues z is therefore solution of:

$$\det\left(I_m-z\tilde{\delta}(z)\delta(z)\mathcal{M}\right)=0$$

where $\mathcal{M}=\lim_{p\to\infty}\frac{1}{c_0}\mathcal{D}_c^{\frac{1}{2}}M^\mathsf{T}M\mathcal{D}_c^{\frac{1}{2}}$. Denoting $\ell_1\geq\ldots\geq\ell_m$ the eigenvalues of $\mathcal{M}$, the eigenvalues $z=\hat{\lambda}_i$ such that $\hat{\lambda}_i>(1+\sqrt{1/c_0})^2$ are explicit and pairwise associated to ℓ_i whenever:

$$\hat{\lambda}_i=\frac{1}{c_0}+1+\ell_i+\frac{1}{c_0\ell_i}>(1+\sqrt{1/c_0})^2$$

which occurs if and only if $\ell_i\geq\frac{1}{\sqrt{c_0}}$. This completes the proof of Proposition 1.

1.2 PCA projectors

In this section, the goal is to study the asymptotic behavior of $u_i^\mathsf{T}\mathbf{x}|\mathbf{x}\in\mathcal{C}_j$, for $i\leq\tau$. Since conditionally on the training data X, $u_i^\mathsf{T}\mathbf{x}$ is expressed as the projection of the deterministic vector u_i on the isotropic gaussian random vector $\mathbf{x}$, it follows that $u_i^\mathsf{T}\mathbf{x}$ is asymptotically Gaussian.

Computation of the mean. Since u_i is independent from $\mathbf{x}$, we have conditionally to the training data X that $\mathbb{E}[u_i^\mathsf{T}\mathbf{x}]=\mu_j^\mathsf{T}u_i$. It then remains to compute the expectation with respect to X. First, since u_i is defined up to a sign, we may impose

$$\mu_j^\mathsf{T}u_i=\frac{\mu_j^\mathsf{T}u_iu_i^\mathsf{T}\mathbb{1}_p/p}{\sqrt{\mathbb{1}_p^\mathsf{T}u_iu_i^\mathsf{T}\mathbb{1}_p/p^2}} \tag{2}$$

Using the Cauchy's integral formula, we have for any vector $a\in\mathbb{R}^p$ of bounded norm (i.e. $\lim_{p\to\infty}\|a\|<\infty$),

$$\begin{aligned}a^\mathsf{T}u_iu_i^\mathsf{T}\frac{\mathbb{1}_p}{p} &= \frac{-1}{2\pi\imath}\oint_{\gamma_i}a^\mathsf{T}\left(\frac{1}{p}WW^\mathsf{T}+\mathcal{U}\mathcal{V}^\mathsf{T}-zI_p\right)^{-1}\frac{\mathbb{1}_p}{p}\\ &= \frac{-1}{2\pi\imath}\oint_{\gamma_i}a^\mathsf{T}\left(Q(z)-Q(z)\mathcal{U}\left(I_{2m}+\mathcal{V}^\mathsf{T}Q(z)\mathcal{U}\right)^{-1}\mathcal{V}^\mathsf{T}Q(z)\right)\frac{\mathbb{1}_p}{p}\\ &= \frac{1}{2\pi\imath}\oint_{\gamma_i}a^\mathsf{T}Q(z)\mathcal{U}\left(I_{2m}+\mathcal{V}^\mathsf{T}Q(z)\mathcal{U}\right)^{-1}\mathcal{V}^\mathsf{T}Q(z)\frac{\mathbb{1}_p}{p}\end{aligned}$$

with γ_i a contour surrounding only the isolated eigenvalues $\hat{\lambda}_i$ of $\frac{1}{p}XX^\mathsf{T}$.

Using the deterministic equivalents of $\tilde{Q}(z)$ and $Q(z)$, we have

$$\begin{aligned}a^\mathsf{T}Q(z)\mathcal{U} &\leftrightarrow \frac{1}{\sqrt{p}}[\delta(z)a^\mathsf{T}M,\mathbb{0}_{1\times m}]\\ I_m+\mathcal{V}^\mathsf{T}Q(z)\mathcal{U} &\leftrightarrow \begin{pmatrix} I_m+\delta(z)\frac{J^\mathsf{T}J}{p}M^\mathsf{T}M & (1+z\tilde{\delta}(z))J^\mathsf{T}J \\ \delta(z)\frac{1}{p}M^\mathsf{T}M & I_m \end{pmatrix}\\ \mathcal{V}^\mathsf{T}Q(z)\frac{\mathbb{1}_p}{p} &\leftrightarrow \frac{1}{\sqrt{p}}\begin{pmatrix}\delta(z)J^\mathsf{T}JM^\mathsf{T}\frac{\mathbb{1}_p}{p}\\ \delta(z)M^\mathsf{T}\frac{\mathbb{1}_p}{p}\end{pmatrix}.\end{aligned}$$

Altogether, this gives :

$$a^\mathsf{T} u_i u_i^\mathsf{T} \frac{\mathbb{1}_p}{p} \leftrightarrow \frac{-1}{2\pi\imath} \oint_{\gamma_i} z\tilde{\delta}(z)\delta(z)^2 a^\mathsf{T} M \mathcal{D}_c^{\frac{1}{2}} \frac{\bar{u}_i \bar{u}_i^\mathsf{T}}{1 - z\delta(z)\tilde{\delta}(z)\ell_i} \mathcal{D}_c^{\frac{1}{2}} M^\mathsf{T} \frac{\mathbb{1}_p}{p} dz$$

with $\bar{u}_i$ the eigenvector of $\mathcal{M}$ associated to the eigenvalue ℓ_i. The only pole of the integrand inside γ_i is the isolated eigenvalue $\hat{\lambda}_i$. From the residue theorem, this gives

$$a^\mathsf{T} u_i u_i^\mathsf{T} \frac{\mathbb{1}_p}{p} \leftrightarrow \frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)} a^\mathsf{T} M \mathcal{D}_c^{\frac{1}{2}} \bar{u}_i \bar{u}_i^\mathsf{T} \mathcal{D}_c^{\frac{1}{2}} M \frac{\mathbb{1}_p}{p}.$$

Finally, using Equation (2), we conclude

$$\mu_j^\mathsf{T} u_i \xrightarrow{\text{a.s.}} \sqrt{\frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)}} \bar{u}_i^\mathsf{T} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} e_j^{[m]}.$$

Computation of the variance. The computation is immediate since U is orthonormal, therefore $\mathrm{Var}(u_i^\mathsf{T}\mathbf{x}) = 1$.

2 Large dimensional analysis of Multi-Task SPCA

We recall that the solution V of SPCA is explicitly given by the collection of the eigenvectors associated with the τ largest eigenvalues of $\frac{1}{p} X \frac{yy^\mathsf{T}}{n} X^\mathsf{T}$. The goal of this section is to evaluate the position of these isolated eigenvalues and to study the behavior of the projection of a new test data on the feature space spanned by SPCA under the large dimensional regime.

Assumption 3 (Distribution of X) *For class j of Task t, denoted $\mathcal{C}_{tj}$, $x_{t\ell}^{(j)} \sim \mathcal{N}(\mu_{tj}, \Sigma_{tj})$, for some $\mu_{tj} \in \mathbb{R}^p$. We further denote $M \equiv [\mu_{11}, \ldots, \mu_{km}] \in \mathbb{R}^{p \times mk}$.*

Assumption 4 (Growth Rate) *As $n \to \infty$, $p/n \to c_0 > 0$ and, for $1 \le j \le m$, $n_{tj}/n \to c_{tj} > 0$; we denote $c = [c_{11}, \ldots, c_{km}]^\mathsf{T} \in \mathbb{R}^{km}$, and $\mathcal{D}_c = \mathrm{diag}(c)$. Besides,*

$$(1/c_0)\mathcal{D}_c^{\frac{1}{2}} M^\mathsf{T} M \mathcal{D}_c^{\frac{1}{2}} \to \mathcal{M} \in \mathbb{R}^{mk \times mk},$$
$$\limsup_p \max\left(\frac{1}{p}\mathrm{tr}\Sigma_{tj}\Sigma_{t'j'}, \frac{1}{p}\mathrm{tr}\Sigma_{tj}\right) < \infty$$

2.1 Isolated eigenvalues

The eigenvalues of $\frac{1}{p} X \frac{yy^\mathsf{T}}{n} X^\mathsf{T}$ are solutions of

$$\det\left(\frac{1}{p} X J \frac{\tilde{y}\tilde{y}^\mathsf{T}}{n} J^\mathsf{T} X^\mathsf{T} - zI_p\right) = \det\left(\frac{1}{p}(\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} \frac{X^\mathsf{T}X}{n} J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} - zI_m\right)$$

Besides we have

$$\frac{1}{n} J^\mathsf{T} \frac{X^\mathsf{T}X}{p} J \leftrightarrow \frac{1}{n} J^\mathsf{T} \mathcal{D}_{\tilde{v}} J + \frac{1}{c_0} \mathcal{D}_c M^\mathsf{T} M \mathcal{D}_c$$

with $\tilde{v} = [\tilde{v}_{11}, \ldots, \tilde{v}_{k2}]$, $\tilde{v}_{tj} = \lim_{p\to\infty} \frac{1}{p}\mathrm{tr}\Sigma_{tj}$.

Therefore, the isolated eigenvalues are, in the large n, p limit, the eigenvalues of $\mathcal{H} = (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(\frac{1}{n} J^\mathsf{T} \mathcal{D}_{\tilde{v}} J + \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{\frac{1}{2}}\right) (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}}$. In the case of identity covariance structure treated in the main article, $\tilde{v}_{tj} = 1,\ \forall t, j$ and therefore

$$\mathcal{H} = (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(\mathcal{D}_c + \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{\frac{1}{2}}\right) (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}}.$$

2.2 SPCA projectors

Computation of the mean. Since the eigenvector v_i is defined up to sign, we may as above impose that

$$\mu_{tj}^\mathsf{T} v_i = \frac{\mu_{tj}^\mathsf{T} v_i v_i^\mathsf{T} \mathbb{1}_p/p}{\sqrt{\mathbb{1}_p^\mathsf{T} v_i v_i^\mathsf{T} \mathbb{1}_p/p^2}}. \tag{3}$$

We have for any vector $a \in \mathbb{R}^p$ such that $\lim_{p\to\infty} \|a\| < \infty$,

$$\begin{aligned}
a^\mathsf{T} v_i v_i^\mathsf{T} \frac{\mathbb{1}_p}{p} &= \frac{-1}{2\pi\imath} \oint_{\gamma_i} a^\mathsf{T} \left(\frac{1}{p} X J \frac{\tilde{y}\tilde{y}^\mathsf{T}}{n} J^\mathsf{T} X^\mathsf{T} - z I_p \right)^{-1} \frac{\mathbb{1}_p}{p} \\
&= \frac{1}{2\pi\imath} \oint_{\gamma_i} \frac{1}{z} a^\mathsf{T} \frac{1}{np} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(z I_m - \frac{1}{np} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \right)^{-1} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} \frac{\mathbb{1}_p}{p} \\
&= \frac{1}{2\pi\imath c_0} \oint_{\gamma_i} \frac{1}{z} a^\mathsf{T} M \mathcal{D}_c (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(z I_m - \mathcal{H} \right)^{-1} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \mathcal{D}_c M^\mathsf{T} \frac{\mathbb{1}_p}{p} + o(1) \\
&= \frac{1}{c_0} \frac{1}{\bar{\lambda}_i} a^\mathsf{T} M \mathcal{D}_c (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \bar{v}_i \bar{v}_i^\mathsf{T} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \mathcal{D}_c M^\mathsf{T} \frac{\mathbb{1}_p}{p} + o(1)
\end{aligned}$$

with γ_i the contour surrounding the eigenvalue $\bar{\lambda}_i$ of $\mathcal{H}$ and $\bar{v}_i$ the eigenvector of $\mathcal{H}$ associated to $\bar{\lambda}_i$.

Therefore,

$$\mu_{tj}^\mathsf{T} v_i \xrightarrow{\text{a.s.}} \sqrt{\frac{1}{\bar{\lambda}_i}} \bar{v}_i^\mathsf{T} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} e_{tj}^{[mk]}.$$

Computation of the variance For the variance, conditionally to the training data X, $\mathrm{Var}(v_i^\mathsf{T}\mathbf{x}) = v_i^\mathsf{T} \Sigma_{tj} v_i$. Furthermore, it then remains to compute the expectation with respect to the training data X:

$$\begin{aligned}
v_i^\mathsf{T} \Sigma_{tj} v_i &= \mathrm{tr}\left(v_i v_i^\mathsf{T} \Sigma_{tj} \right) \\
&= \frac{-1}{2\pi\imath} \mathrm{tr} \left(\Sigma_{tj} \oint_{\gamma_i} \left(\frac{1}{p} X J \frac{\tilde{y}\tilde{y}^\mathsf{T}}{n} J^\mathsf{T} X^\mathsf{T} - z I_p \right)^{-1} \right) \\
&= \frac{1}{2\pi\imath} \mathrm{tr} \left(\Sigma_{tj} \oint_{\gamma_i} \frac{1}{npz} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(z I_m - \frac{1}{np} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \right)^{-1} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} \right) \\
&= \frac{1}{2\pi\imath} \mathrm{tr} \left(\oint_{\gamma_i} \frac{1}{npz} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} \Sigma_{tj} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \left(z I_m - \frac{1}{np} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} J^\mathsf{T} X^\mathsf{T} X J (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \right)^{-1} \right) \\
&= \frac{1}{\bar{\lambda}_i} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} \mathcal{T}_{tj} (\tilde{y}\tilde{y}^\mathsf{T})^{\frac{1}{2}} + o(1)
\end{aligned}$$

where $\mathcal{T}_{tj} = \frac{1}{n} J^\mathsf{T} \mathcal{D}_{\bar{v}} J + \mathcal{D}_c^{\frac{1}{2}} \mathcal{M} \mathcal{D}_c^{\frac{1}{2}}$ and $\bar{v}_{ab} = \lim_{p\to\infty} \frac{1}{p} \mathrm{tr}\left(\Sigma_{tj} \Sigma_{ab} \right)$.

When $\Sigma_{tj} = I_p$, as treated in the main article, it is immediate that $\mathrm{Var}(u_i^\mathsf{T}\mathbf{x}) = 1$.

3 Binary-based multi-class classification

This section provides various applications and optimizations of the proposed MTL-SPCA framework in the context of multi-class classification.

3.1 One-versus-all multi-class preliminary

The literature [3] describes broad groups of approaches to deal with classification with $m > 2$ classes. We focus here on the most common method, namely the one-versus-all approach. The complete optimization of one-versus-all being theoretically heavy to handle and demanding prior knowledge on the decision output statistics, the method inherently suffers from sometimes severe practical

limitations; these are partly tackled here exploiting the large dimensional analysis performed in this article.

In the one-versus-all method, focusing on Task t, m individual binary classifiers, indexed by $\ell = 1, \ldots, m$, are trained, each of them separating Class $\mathcal{C}_{t\ell}$ from the other $m-1$ classes $\mathcal{C}_{t\ell'}$, $\ell' \neq \ell$. Each test sample is then allocated to the class index corresponding to the classifier reaching the highest of the m classifier scores. Although quite used in practice, the approach first suffers a severe unbalanced data bias 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 center-scale issue when ultimately comparing the outputs of the m decision functions, the average locations and ranges of which may greatly differ; these issues lead to undesirable effects, as reported in [3, section 7.1.3]).

These problems are here simultaneously addressed: specifically, having access to the large dimensional statistics of the classification scores allows us to appropriately center and scale the scores. Each centered-scaled binary classifier is then further optimized by appropriately selecting the class labels (different from ± 1) so to minimize the resulting classification error. See Figure 1 for a convenient illustration of the improvement induced by this centering-scaling and label optimization approach.

3.2 One-versus-all multi-class optimization

For each target task t, in a one-to-all approach, m MTL-SPCA binary classifications are solved with the target class $\mathcal{C}_{t\ell}$ (renamed "class $\mathcal{C}_{t1}^{\ell}$"), against all other $\mathcal{C}_{t2}^{\ell}$ classes (combined into a single "$\mathcal{C}_{t2}^{\ell}$ class"). Calling $g_{\mathbf{x},t}^{(\ell)}$ the output of the classifier ℓ for a new datum $\mathbf{x}$ in Task t, the class allocation decision is traditionally based on the largest among all scores $g_{\mathbf{x},t}^{(1)}, \ldots, g_{\mathbf{x},t}^{(m)}$. However, this presumes that the distribution of the scores $g_{\mathbf{x},t}^{(1)}$ when $\mathbf{x} \in \mathcal{C}_1$, $g_{\mathbf{x},t}^{(2)}$ when $\mathbf{x} \in \mathcal{C}_2$, etc., more or less have the same statistical mean and variance. This is not the case in general, as depicted in the first column of Figure 1, where data from class $\mathcal{C}_1$ are more likely to be allocated to class $\mathcal{C}_3$ (compare the red curves).

By providing an accurate estimate of the distribution of the scores $g_{\mathbf{x},t}^{(\ell)}$ for all ℓ's and all genuine classes of $\mathbf{x}$, Theorem 3 of the main article allows us to predict the various positions of the Gaussian curves in Figure 1. In particular, it is possible, for each binary classifier ℓ to center and scale $g_{\mathbf{x},t}^{(\ell)}$ when $\mathbf{x} \in \mathcal{C}_{t\ell}$. This operation averts the centering and scaling biases depicted in the first column of Figure 1: the result of the center-scale operation appears in the second column of Figure 1.

This first improvement step simplifies the algorithm which now boils down to determining the index of the largest $g_{\mathbf{x},t}^{(\ell)} - m_{t1}^{(bin,\ell)}$, $\ell \in \{1, \ldots, m\}$, while limiting the risks induced by the center-scale biases.

This being said, our theoretical analysis further allows to adapt the input labels $\tilde{y}_{[t]}^{(\ell)}$ in such a way to optimize the expected output. Ideally, assuming $\mathbf{x}$ genuinely belongs to class $\mathcal{C}_{t\ell}$, one may aim to increase the distance between the output score $g_{\mathbf{x},t}^{(\ell)}$ and the other output scores $g_{\mathbf{x},t}^{(\ell')}$ for $\ell' \neq \ell$. This however raises two technical questions:

1. Corollary 1 of the main article is derived under a 2-class Gaussian mixture model while for classifier ℓ of the one-versus-all approach, the data are composed of m Gaussians, of which one belongs to class $\mathcal{C}_{t1}^{\ell}$ and the other $m-1$ to class $\mathcal{C}_{t2}^{\ell}$ (which remains a mixture when $m > 2$). In this case, the labels express as $y = J\tilde{y}$, with now $\tilde{y} \in \mathbb{R}^{mk}$ (instead of $\mathbb{R}^{2k}$) for $J = \begin{pmatrix} \mathbb{1}_{n_{11}} & & \\ & \ddots & \\ & & \mathbb{1}_{n_{mk}} \end{pmatrix}$;
2. the procedure demands to simultaneously adapt all input scores $\tilde{y}_{[t]}^{(1)}, \ldots, \tilde{y}_{[t]}^{(m)}$.

To solve Item 1., we extend Corollary 1 to a one-versus-all based binary classification.

Corollary 1 (One-versus-all Binary MTL Supervised Principal Component Analysis) *Let* $\mathbf{x} \sim \mathcal{N}(\mu_{tj}, I_p)$ *independent of* X*. Then, under Assumptions 3-4 and the above setting, as* $p, n \to \infty$*,*

$$V^\mathsf{T}\mathbf{x} - G_{tj} \to 0, \quad G_{tj} \sim \mathcal{N}(\mathfrak{m}_{tj}^{(\mathrm{bin})}, 1), \quad \textit{where} \quad \mathfrak{m}_{tj}^{(\mathrm{bin})} = \frac{\tilde{y}^\mathsf{T}\mathcal{D}_c^{\frac{1}{2}}\mathcal{M}\mathcal{D}_c^{-\frac{1}{2}}e_{tj}}{\sqrt{\tilde{y}^\mathsf{T}(\mathcal{D}_c^{\frac{1}{2}}\mathcal{M}\mathcal{D}_c^{\frac{1}{2}} + \mathcal{D}_c)\tilde{y}}}.$$

Note that Corollary 1 is similar to Corollary 1 of the main article but now with $\tilde{y} \in \mathbb{R}^{mk}$ and $\mathcal{M}, \mathcal{D}_c \in \mathbb{R}^{mk \times mk}$.

A first option to solve Item 2. consists in maximizing the distance between the output score $g_{\mathbf{x},t}^{(\ell)}$ for $\mathbf{x} \in \mathcal{C}_{t\ell}$ and the scores $g_{\mathbf{x},t}^{(\ell)}$ for $\mathbf{x} \notin \mathcal{C}_{t\ell}$. By "mechanically" pushing away all wrong decisions, this ensures that, when $\mathbf{x} \in \mathcal{C}_{t\ell}$, $g_{\mathbf{x},t}^{(\ell)}$ is greater than $g_{\mathbf{x},t}^{(\ell')}$ 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, the proposed (heuristic) label "optimization" here consists in solving, for a target Task t and each $\ell \in \{1, \ldots, m\}$ the optimization problem:

$$\tilde{y}_{[t]}^{\star(\ell)} = \max_{\tilde{y}_{[t]}^{(\ell)} \in \mathbb{R}^{km}} \min_{j \neq \ell} \left(\mathfrak{m}_{t\ell}^{(\mathrm{bin}),\ell} - \mathfrak{m}_{tj}^{(\mathrm{bin}),\ell} \right) \tag{4}$$

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

Being a non-convex and non-differentiable (due to the max) optimization, Equation (4) cannot be solved straightforwardly. An approximated solution consists in relaxing the max operator $\max(x_1, \ldots, x_n)$ into the differentiable soft-max 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 $\tilde{y}_{[t]}^{(\ell)} \in \mathbb{R}^{mk}$ filled with 1's at every $m(i'-1)+\ell$, for $i' \in \{1, \ldots, m\}$, and with -1's everywhere else.

An alternative option to tackle Item 2. (the one developed in the core article) consists in reducing the dimension of the labels to $\tilde{y}_{[t]}^{(\ell)} \in \mathbb{R}^{2k}$ by "merging" all Gaussians of class $\mathcal{C}_{tj}$ with $j \neq \ell$ into a unique *approximated* Gaussian class with mean $\sum_{j \neq \ell} \frac{n_{tj}}{n - n_{t\ell}} \mu_{tj}$. We may then (abusively) apply Corollary 1, leading to an explicit expression of the optimal label $\tilde{y}_{[t]}^{\star(\ell)}$, from which Algorithm 1 in the main article unfolds.

Figure 2 compares the "Min-Max" optimization scheme with the scheme assuming the Gaussian approximation for class 2 (denoted "Gaussian Approx"). The two methods interestingly have comparable performance. The synthetic data considered for this experiment consists of 2-tasks with ten Gaussian classes with means $\mu_{1j} = \mu_j$ and $\mu_{2j} = \beta\mu_j + \sqrt{1-\beta^2}\mu_j^\perp$.

4 Supplementary experiments

We next experiment on two transfer learning datasets:

- the Office31 dataset [6] which contains 31 object categories in three domains: Amazon (A), DSLR (D) and Webcam (W). The Amazon images were captured from a website of online merchants (clean background and unified scale). The DSLR domain contains low-noise high resolution images. For Webcam, the images of low resolution exhibit significant noise and color. Every pair of domains is successively selected as "source" and a "target" for binary (transfer) multi-task learning, resulting in 6 transfer tasks S→T for S,T∈ {A,D,W};
- the OfficeHome dataset [7] which consists of images from 4 different domains: Artistic images (A), Clip Art (C), Product images (P) and Real-World images (R). For each domain, the dataset contains images of 65 object categories found typically in Office and Home settings.

Table 1 reports the comparative performances of the various algorithms and, while exhibiting a slight superiority for the MTL-LSSVM scheme, supports the stable and competitive performance of MTL-SPCA.

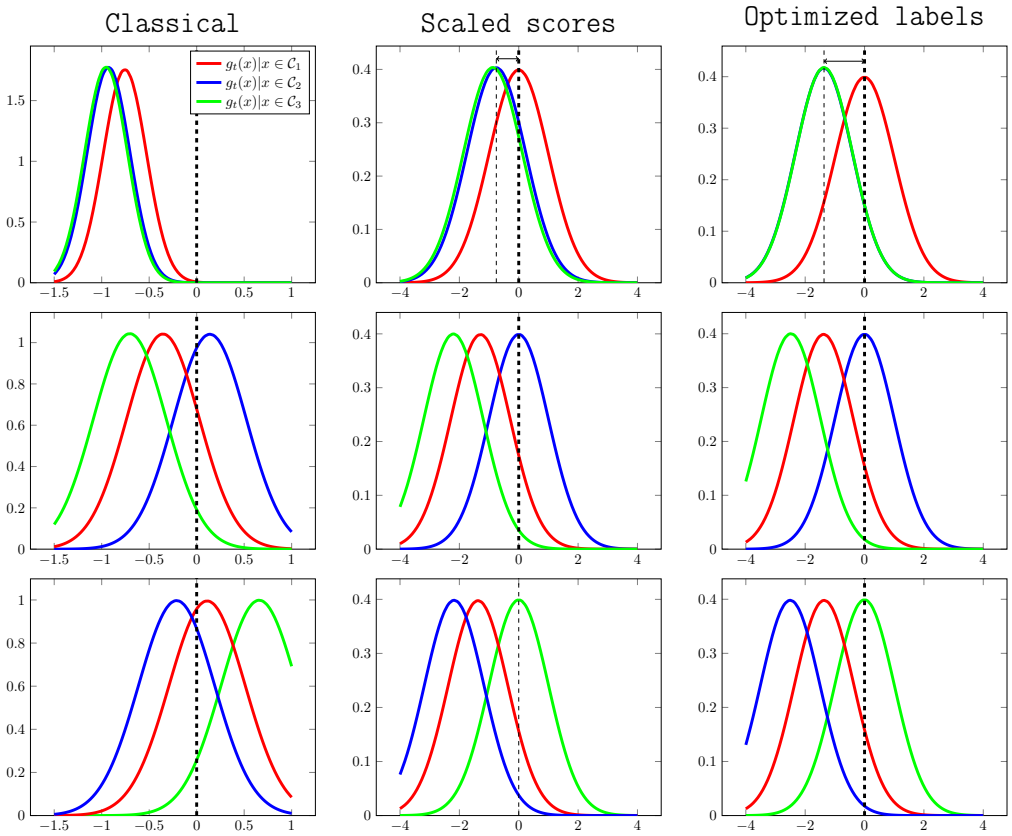


Figure 1: Test score distribution in a 2-task and 3 classes-per-task setting, using a one-versus-all multi-class classification. Every graph in row ℓ depicts the limiting distributions of $g^{(\ell)}_{\mathbf{x},t}$ for $\mathbf{x}$ in different classes. Column 1 (Classical) is the standard implementation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

S/T	w → a	w → d	a → w	a → d	d → w	d → a	Mean score
ST-SPCA	77.63	93.72	90.09	90.51	91.33	75.43	86.45
CDLS	76.47	92.52	91.57	90.07	91.43	74.99	86.17
N-SPCA	74.10	96.44	79.59	81.94	95.10	73.15	83.39
MTL-LSSVM	**80.85**	**97.63**	**93.11**	**91.91**	**95.12**	**79.41**	**89.67**
MTL SPCA	*77.67*	*96.70*	*90.72*	*91.09*	*94.83*	*76.90*	*87.99*

Table 1: Classification accuracy over Office31 database. w(Webcam), a(Amazon), d(dslr), for different "Source to target" task pairs ($S \to T$) based on Resnet-50 features.

References

[1] Jinho Baik and Jack W Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of multivariate analysis*, 97(6):1382–1408, 2006.

[2] Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

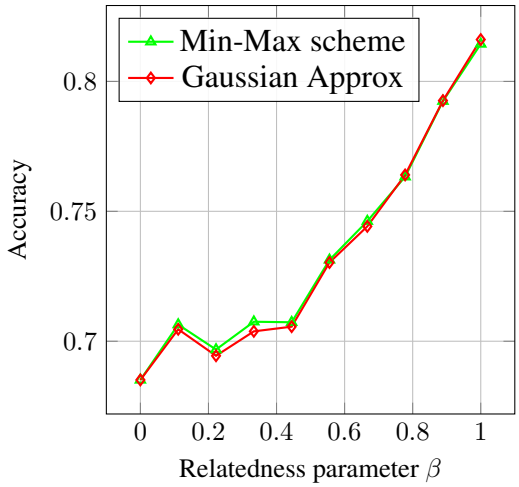


Figure 2: Empirical accuracy as function of the relatedness parameter β on Synthetic Gaussian with $p = 500$, $\mu_j = 3e_j^{[p]}$, $\mu_j^{\perp} = 3e_{p-j}^{[p]}$, $n_{1j} = 100$, $n_{2j} = 50$ for $1 \leq j \leq 10$; 10 000 test sample averaging

S/T	A → R	A → P	A → C	R → A	R → P	R → C	P → A	P → R	P → C	C → A	C → R	C → P	Mean score
ST-SPCA	91.07	92.19	74.05	77.61	92.64	72.84	75.66	90.38	71.48	72.26	86.47	89.20	82.15
CDLS	88.30	90.24	75.71	78.04	91.28	*75.29*	75.59	88.20	73.86	73.43	85.12	88.91	82.00
N-SPCA	89.73	89.26	69.47	76.77	89.90	66.63	71.13	87.41	63.01	70.50	84.30	82.98	78.42
MTL LSSVM	**91.82**	**92.85**	**80.09**	**79.39**	**93.63**	**79.13**	*75.94*	**90.67**	**78.19**	**74.39**	**88.61**	**91.56**	**84.69**
MTL SPCA	*91.10*	*92.28*	*77.44*	*79.57*	*92.79*	73.64	**76.36**	*90.39*	*76.90*	*74.23*	*87.01*	*89.37*	*83.42*

Table 2: Classification accuracy over Office+Home database. Art (A), RealWorld (R), Product (P), Clipart (C), for different "Source to target" task pairs ($S \to T$) based on Resnet-50 features.

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

[4] Romain Couillet and Merouane Debbah. *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[5] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

[6] 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.

[7] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.