PCA-based Multi-Task Learning: a Random Matrix Approach

Malik Tiomoko [1] **Romain Couillet** [2] **Frédéric Pascal** [3]

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 (Barshan et al., 2011; Bair et al., 2006). 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 is often too demanding in practice. Multi-task learning (MTL) (Caruana, 1997; Zhang & Yang, 2018; 2021) 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 theoretically identifying the common features of the datasets. Several heuristics have been proposed, which may be split into two groups: parameter- versus feature-based MTL. In parameter-based MTL, the tasks are assumed to share common hyperparameters (Evgeniou & Pontil, 2004; Xu et al., 2013) (*e.g.*, separating hyperplanes in a support vector machine (SVM) flavor) or hyperparameters derived from a common prior distribution (Zhang & Yeung, 2012; 2014). Classical learning mechanisms (SVM, logistic regression, etc.) can be appropriately turned into an MTL version by enforcing parameter relatedness: (Evgeniou & Pontil, 2004; Xu et al., 2013; Parameswaran & Weinberger, 2010) 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 to be identified: through sparse coding, deep neural network embeddings, principal component analysis (PCA) (Argyriou et al., 2008; Maurer et al., 2013; Zhang et al., 2016; Pan et al., 2010) or simply by feature selection (Obozinski et al., 2006; Wang & Ye, 2015; Gong et al., 2012).

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 (Rosenstein et al., 2005; Long et al., 2013); this often occurs when task relatedness is weaker than assumed, and MTL enforces fictitious similarities.

A large dimensional analysis to improve MTL. To fill these gaps, this work focuses on a large dimensional random matrix setting (El Karoui, 2018) to provide an exact (asymptotic) performance evaluation of an elementary (yet powerful) PCA-based MTL approach. It is worth noticing that although the proposed framework is asymptotic, it has shown to be very efficient for small dimensions/small numbers of data in practice (see *e.g.* (Couillet & Liao, 2022)). This analysis conveys insights into the MTL inner workings, providing an optimal data labelling scheme to avert negative transfer fully.

More fundamentally, the choice of investigating PCA-based

[1]Huawei Noah's Ark Lab, Paris, France [2]LIG-Lab, Université de Grenoble Alpes, France [3]L2S Centrale-Supélec, France. Correspondence to: Malik Tiomoko <malik.tiomoko@huawei.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

MTL results from realizing that the potential gains incurred by a proper theoretical adaptation of simple algorithms vastly 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 article's main contribution 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 (Lacoste et al., 2019; Strubell et al., 2019; Henderson et al., 2020).

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 by-product result, this work formally provides the optimal choice of dimension for PCA and 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) theoretically 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 before running the SPCA-MTL algorithm;
- Synthetic and real data experiments support the competitive SPCA-MTL results 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. p stands for the data dimension while n corresponds to the data number, m is the number of classes and k stands for the tasks. Vectors (resp. matrices) are denoted by bold-faced lowercase letters (resp. uppercase letters). $\mathbf{e}_m^{[n]} \in \mathbb{R}^n$ is the canonical vector of $\mathbb{R}^n$ with $[\mathbf{e}_m^{[n]}]_i = \delta_{mi}$. Moreover, $\mathbf{e}_{ij}^{[mk]} = \mathbf{e}_{m(i-1)+j}^{[mk]}$.

2. Related works

A series of supervised (single-task) learning methods were proposed which rely on PCA (Barshan et al., 2011; Ritchie et al., 2019; Zhang et al., 2021; Ghojogh & Crowley, 2019): 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 understanding the statistics of the PCA eigenvectors: only recently have large dimensional statistics, and specifically random matrix theory, provided the first insights into the behavior of eigenvalues and eigenvectors of sample covariance and kernel matrices (Benaych-Georges & Nadakuditi, 2012; Johnstone, 2001; Baik & Silverstein, 2006; Lee et al., 2010; Paul, 2007). To the best of our knowledge, none of these works have drawn an analysis of SPCA: the closest work is likely (Ashtiani & Ghodsi, 2015) which however only provides statistical bounds on performance rather than exact results.

On the MTL side, several methods were proposed under unsupervised (Long et al., 2016; Saito et al., 2018; Baktashmotlagh et al., 2013), semi-supervised (Rei, 2017; Liu et al., 2007) and supervised (parameter-based (Tiomoko et al., 2020; Evgeniou & Pontil, 2004; Xu et al., 2013; Ando & Zhang, 2005) or feature-based (Argyriou et al., 2008; Liu et al., 2012)) flavors. Although most of these works generally achieve satisfying performances on both synthetic and real data, few theoretical analyses and guarantees exist so 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 (Ruder, 2017). These are, however, at odds with the article's 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 those 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 $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ of n samples of dimension

[1] But nothing prevents us from exploiting data features extracted from pre-trained deep nets.

p are needed. To each $\mathbf{x}_i \in \mathbb{R}^p$ is attached a (possibly multivariate) label $\mathbf{y}_i$: in a binary class setting, $y_i \in \{-1, 1\}$, while for $m \geq 3$ classes, $\mathbf{y}_i = \mathbf{e}_j^{[m]} \in \mathbb{R}^m$, the canonical vector of the corresponding class j.

PCA in supervised learning. Let us first recall that, applied to $\mathbf{X}$, PCA identifies a subspace of $\mathbb{R}^p$, say the span of the columns of $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_\tau] \in \mathbb{R}^{p \times \tau}$ ($\tau \leq p$), which maximizes the variance of the data when projected on the subspace, i.e., $\mathbf{U}$ solves:

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

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

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 $\mathbf{U}$ and in performing classification in the projected space. This has the strong advantage of providing a (possibly dramatic) dimensionality reduction (from p to τ) to supervised learning mechanisms, thus improving cost efficiency while mitigating the loss incurred by the dimension reduction. Yet, the PCA step is fully unsupervised and does not exploit the available class information. It is instead proposed in (Barshan et al., 2011; Chao et al., 2019) to trade $\mathbf{U}$ for a more representative projector $\mathbf{V}$ which "maximizes the dependence" between the projected data $\mathbf{V}^\mathsf{T}\mathbf{X}$ and the output labels $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\mathsf{T} \in \mathbb{R}^{n \times m}$. To this end, (Barshan et al., 2011) exploits the Hilbert-Schmidt independence criterion (Gretton et al., 2005), with corresponding optimization

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

This results in the *Supervised PCA* (SPCA) projector, the solution $\mathbf{V} = \mathbf{V}(\mathbf{Y})$ of which being the concatenation of the τ dominant eigenvectors of $\frac{\mathbf{X}\mathbf{Y}\mathbf{Y}^\mathsf{T}\mathbf{X}^\mathsf{T}}{np}$. Subsequent learning (by SVMs, empirical risk minimizers, discriminant analysis, etc.) is then applied to the projected training $\mathbf{V}^\mathsf{T}\mathbf{x}_i$ and test $\mathbf{V}^\mathsf{T}\mathbf{x}$ data.

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 3.1 (Distribution of $\mathbf{X}$)**.** The columns of $\mathbf{X}$ are independent random vectors with $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_m]$, $\mathbf{X}_j = [\mathbf{x}_1^{(j)}, \ldots, \mathbf{x}_{n_j}^{(j)}] \in \mathbb{R}^{p \times n_j}$ for $\mathbf{x}_i^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{I}_p)$, also denoted $\mathbf{x}_i^{(j)} \in \mathcal{C}_j$ and $\mathbf{M} \equiv [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m] \in \mathbb{R}^{p \times m}$. Note that $\sum_{j=1}^m n_j = n$.

Recent works in random matrix (Seddik et al., 2020; Louart & Couillet, 2018) suggests that the technical arguments used in this paper are extendable to the broader family of random vectors known as concentrated random vectors in which a wide range of realistic random vectors can be found, including Generative Adversarial Network images. Moreover, the experiments on image and text classification suggest the robustness of the intuitions drawn on real data.

Assumption 3.2 (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 $\mathbf{c} = [c_1, \ldots, c_m]^\mathsf{T}$ and $D_\mathbf{c} = \operatorname{diag}(\mathbf{c})$. Besides,

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

The growth rate assumption assumes that p and n are of the same order of magnitude. This is different and more realistic than assumptions usually considered in classical statistics, where n is very large compared to p. More importantly, the technical results are obtained at a convergence rate of order of $1/\sqrt{p}$, which allows a smooth application to finite p, n. The assumption on the existence of the matrix $\mathcal{M}$ states the difficulty of the problem when p evolves. Any other rate for the order of $\mathcal{M}$ (scaling with p for example) will lead to a trivial problem (tasks too easy or too difficult to solve).

Let us now show that, under this setting, SPCA is uniformly more discriminative on new data than PCA. As $n, p \to \infty$, the spectrum of $\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T}$ is subject to a *phase transition phenomenon* now well established in random matrix theory (Baik & Silverstein, 2006; Benaych-Georges & Nadakuditi, 2012). This result is crucial as the PCA vectors of $\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T}$ are *only informative* beyond the phase transition and otherwise can be considered pure noise.

Proposition 3.3 (Eigenvalue Phase transition)**.** *Under Assumptions 3.1-3.2, 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 $\frac{\mathbf{X}\mathbf{X}^\mathsf{T}}{p}$ converges weakly, with probability one, to the Marčenko-Pastur law (Marchenko & Pastur, 1967) 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]

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

$$\lambda_{m+1} \xrightarrow{\text{a.s.}} (1 + \sqrt{1/c_0})^2.$$

[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.

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

Proposition 3.3 states that, if $\ell_i \geq 1/\sqrt{c_0}$, the i-th largest eigenvalue of $\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T}$ 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 $\boldsymbol{\mu}_j$'s). Consequently, $\mathbf{U}^\mathsf{T}\mathbf{x} \in \mathbb{R}^\tau$ also exhibits a phase transition phenomenon.

Theorem 3.4 (Asymptotic behavior of PCA projectors)**.** *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_j, I_p)$ independent of $\mathbf{X}$. Then, under Assumptions 3.1-3.2, with $(\ell_i, \bar{\mathbf{u}}_i)$ the decreasing (distinct) eigenpairs of $\mathcal{M}$, as $p, n \to \infty$,*

$$\mathbf{U}^\mathsf{T}\mathbf{x} - \mathbf{G}_j \to 0, \quad \mathbf{G}_j \sim \mathcal{N}(\mathfrak{m}_j^{(\mathrm{pca})}, \mathbf{I}_\tau), \quad \textit{in probability,}$$

where $[\mathfrak{m}_j^{(\mathrm{pca})}]_i =$

$$\begin{cases} \sqrt{\frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)}}\bar{\mathbf{u}}_i^\mathsf{T}\mathcal{M}\mathrm{D}_\mathbf{c}^{-\frac{1}{2}}\mathbf{e}_j^{[m]}, & i \leq \min(m,\tau) \text{ and } \ell_i \geq \frac{1}{\sqrt{c_0}} \\ 0, & \text{otherwise.} \end{cases}$$

As such, only the projections on the eigenvectors of $\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T}$ 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 $\frac{\mathbf{X}\mathbf{Y}\mathbf{Y}^\mathsf{T}\mathbf{X}^\mathsf{T}}{np}$ only has m non-zero eigenvalues, no phase transition occurs: all eigenvalues are "isolated". Thus, one may take $\tau = m$ principal eigenvectors for the SPCA projection matrix $\mathbf{V}$, which is quite likely informative.

Theorem 3.5 (Asymptotic behavior of SPCA projectors)**.** *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{I}_p)$ independent of $\mathbf{X}$. Then, under Assumptions 3.1-3.2, as $p, n \to \infty$, in probability,*

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

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

Since both PCA and SPCA data projections $\mathbf{U}^\mathsf{T}\mathbf{x}$ and $\mathbf{V}^\mathsf{T}\mathbf{x}$ 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). 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).

Remark 3.6 (Consistent estimate of sufficient statistics). From Assumption 3.2, c_j can be empirically estimated by n_j/n. This, in turn, provides a consistent estimate for $D_\mathbf{c}$. Besides, as $n, p \to \infty$,

$$\frac{1}{n_j n_{j'}}\mathbb{1}_{n_j}^\mathsf{T}\mathbf{X}_j^\mathsf{T}\mathbf{X}_{j'}\mathbb{1}_{n_{j'}} \xrightarrow{\text{a.s.}} [\mathbf{M}^\mathsf{T}\mathbf{M}]_{jj'},\ \forall j \neq j' \quad \text{and}$$
$$\frac{4}{n_j^2}\mathbb{1}_{\frac{n_j}{2}}^\mathsf{T}\mathbf{X}_{j,1}^\mathsf{T}\mathbf{X}_{j,2}\mathbb{1}_{\frac{n_j}{2}} \xrightarrow{\text{a.s.}} [\mathbf{M}^\mathsf{T}\mathbf{M}]_{jj},\ \forall j$$

where $\mathbf{X}_j = [\mathbf{X}_{j,1}, \mathbf{X}_{j,2}] \in \mathbb{R}^{p\times n_j}$, with $\mathbf{X}_{j,1}, \mathbf{X}_{j,2} \in \mathbb{R}^{p\times(n_j/2)}$. 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 $\mathbf{x}$:

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

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

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

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

$$\Delta\mathfrak{m}_{(j,j')}^{(\mathrm{spca})} - \Delta\mathfrak{m}_{(j,j')}^{(\mathrm{pca})} = \sum_{i=1}^{\tau}\frac{\left(\bar{\mathbf{v}}_i^\mathsf{T}\mathcal{M}D_\mathbf{c}^{-\frac{1}{2}}(\mathbf{e}_j^{[\tau]} - \mathbf{e}_{j'}^{[\tau]})\right)^2}{\ell_i^2(\ell_i+1)} \geq 0.$$

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

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

where $\Delta\boldsymbol{\mu} \equiv \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, conveniently showing the influence of n/p and of $\|\Delta\boldsymbol{\mu}\|^2$ in the relative performance gap, which vanishes as the task gets easier or as n/p increases (so with more data).

In summarizing, under a large dimensional setting, we showed that SPCA-based classification uniformly outperforms the PCA alternative, thus motivating the design of

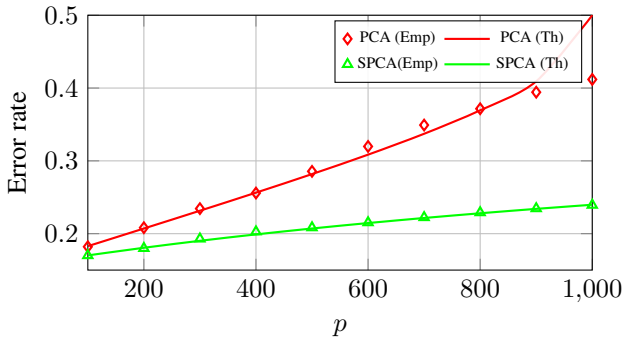


Figure 1. Theoretical (Th) vs. empirical (Emp) error for PCA- and SPCA-based binary classification: $\mathbf{x}_i^{(\ell)} \sim \mathcal{N}((-1)^\ell \boldsymbol{\mu}, \mathbf{I}_p)$ ($\ell \in \{1,2\}$), $\boldsymbol{\mu} = \mathbf{e}_1^{[p]}$, $n_1 = n_2 = 500$. Averaged over 1 000 test samples.

an SPCA-based MTL approach. Furthermore, the section not only justifies the superiority of SPCA over PCA qualitatively but, more importantly quantitatively by quantifying the gap and highlighting the elements that influence it (the norm of the means of the data $\|\boldsymbol{\mu}\|^2$ and the ratio $\frac{p}{n}$ notably).

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

4.1. Multi-task setting

Let now $\mathbf{X} = [\mathbf{X}_{[1]}, \ldots, \mathbf{X}_{[k]}] \in \mathbb{R}^{p \times n}$ 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 $\mathbf{X}_{[t]} = [\mathbf{X}_{[t]1}, \ldots, \mathbf{X}_{[t]m}] \in \mathbb{R}^{p \times n_t}$ with $\mathbf{X}_{[t]j} = [\mathbf{x}_{t1}^{(j)}, \ldots, \mathbf{x}_{tn_{tj}}^{(j)}] \in \mathbb{R}^{p \times n_{tj}}$ 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 $\mathbf{x}_{t\ell}^{(j)} \in \mathbb{R}^p$ is attached a corresponding "label" (or score) $\mathbf{y}_{t\ell}^{(j)} \in \mathbb{R}^m$. We denote in short $\mathbf{Y}_t = [\mathbf{y}_{t1}^{(1)}, \ldots, \mathbf{y}_{tn_t}^{(m)}]^\mathsf{T} \in \mathbb{R}^{n_t \times m}$ and $\mathbf{Y} = [\mathbf{Y}_1^\mathsf{T}, \ldots, \mathbf{Y}_k^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{n \times m}$ the matrix of all labels. The natural MTL extension of SPCA would default $\mathbf{y}_{t\ell}^{(j)} \in \mathbb{R}^m$ to the canonical vectors $\mathbf{e}_j^{[m]}$ (or to ± 1 in the binary case). We disrupt here from this approach by explicitly *not* imposing a value for $\mathbf{y}_{t\ell}^{(j)}$: this will be seen to be key to *avert the problem of negative transfer*. We only let $\mathbf{y}_{t\ell}^{(j)} = \tilde{\mathbf{y}}_{tj}$, for all $1 \le \ell \le n_{tj}$ and for some generic matrix $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_{11}, \ldots, \tilde{\mathbf{y}}_{km}]^\mathsf{T} \in \mathbb{R}^{mk \times m}$, i.e., we impose that

$$\mathbf{Y} = \mathbf{J}\tilde{\mathbf{Y}}, \quad \text{for} \quad \mathbf{J} = [\mathbf{j}_{11}, \ldots, \mathbf{j}_{mk}],$$

$$\text{where} \quad \mathbf{j}_{tj} = (0, \ldots, 0, \mathbb{1}_{n_{tj}}, 0, \ldots, 0)^\mathsf{T}.$$

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

Assumption 4.1 (Distribution of $\mathbf{X}$)**.** For class j of task t, denoted $\mathcal{C}_{tj}$, $\mathbf{x}_{t\ell}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}_{tj}, \mathbf{I}_p)$, for some $\boldsymbol{\mu}_{tj} \in \mathbb{R}^p$. We further denote $\mathbf{M} \equiv [\boldsymbol{\mu}_{11}, \ldots, \boldsymbol{\mu}_{km}] \in \mathbb{R}^{p \times mk}$.

Assumption 4.2 (Growth Rate)**.** As $n \to \infty$, $p/n \to c_0 > 0$ and, for $1 \le j \le m$, $n_{tj}/n \to c_{tj} > 0$. Denoting $\mathbf{c} = [c_{11}, \ldots, c_{km}]^\mathsf{T} \in \mathbb{R}^{km}$ and $D_\mathbf{c} = \mathrm{diag}(\mathbf{c})$, $(1/c_0) D_\mathbf{c}^{\frac{1}{2}} \mathbf{M}^\mathsf{T} \mathbf{M} D_\mathbf{c}^{\frac{1}{2}} \to \mathcal{M} \in \mathbb{R}^{mk \times mk}$.

We are now able to present the article's main technical result.

Theorem 4.3 (MTL Supervised Principal Component Analysis)**.** *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{tj}, \mathbf{I}_p)$ independent of $\mathbf{X}$ and $\mathbf{V} \in \mathbb{R}^{p \times \tau}$ be the collection of the $\tau \le mk$ dominant eigenvectors of $\frac{\mathbf{X}\mathbf{Y}\mathbf{Y}^\mathsf{T}\mathbf{X}^\mathsf{T}}{np} \in \mathbb{R}^{p \times p}$. Then, under Assumptions 4.1-4.2, as $p, n \to \infty$, in probability,*

$$\mathbf{V}^\mathsf{T}\mathbf{x} - \mathbf{g}_{tj} \to 0, \quad \mathbf{g}_{tj} \sim \mathcal{N}(\mathfrak{m}_{tj}, \mathbf{I}_\tau)$$

$$\text{for} \quad [\mathfrak{m}_{tj}]_i = \sqrt{1/(c_0 \tilde{\ell}_i)}\, \bar{\mathbf{v}}_i^\mathsf{T} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \mathrm{D}_\mathbf{c}^{\frac{1}{2}} \mathcal{M} \mathrm{D}_\mathbf{c}^{-\frac{1}{2}} \mathbf{e}_{tj}^{[mk]}$$

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

As in the single task case, despite the high dimension of the data statistics appearing in $\mathbf{V}$, the asymptotic performance only depends on the (small) $mk \times mk$ matrices $\mathcal{M}$ and $D_\mathbf{c}$, which here leverages the inter-task inter-class products $\boldsymbol{\mu}_{tj}^\mathsf{T} \boldsymbol{\mu}_{t'j'}$. This correlation between tasks *together with the labelling choice* $\tilde{\mathbf{Y}}$ (importantly recall that here $\mathbf{V} = \mathbf{V}(\mathbf{Y})$) influences the MTL performance. The next section discusses how to optimally *align* $\tilde{\mathbf{Y}}$ *and* $\mathcal{M}$ so to maximize this performance. This, in addition to Remark 3.6 being still valid here (i.e., $\mathbf{c}$ 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

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

Corollary 4.4 (Binary MTL Supervised Principal Component Analysis)**.** *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{tj}, I_p)$ independent of $\mathbf{X}$. Then, under Assumptions 4.1-4.2 and the above setting, as*

[4] For simplicity, 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.

$p, n \to \infty$,

$$\mathbf{v}^\mathsf{T}\mathbf{x} - g_{tj} \to 0, \quad g_{tj} \sim \mathcal{N}(m_{tj}^{(\mathrm{bin})}, 1)$$

$$\textit{where} \quad m_{tj}^{(\mathrm{bin})} = \frac{\tilde{\mathbf{y}}^\mathsf{T} \mathrm{D}_\mathbf{c}^{\frac{1}{2}} \mathcal{M} \mathrm{D}_\mathbf{c}^{-\frac{1}{2}} \mathbf{e}_{tj}}{\sqrt{\tilde{\mathbf{y}}^\mathsf{T}(\mathrm{D}_\mathbf{c}^{\frac{1}{2}} \mathcal{M} \mathrm{D}_\mathbf{c}^{\frac{1}{2}} + \mathrm{D}_\mathbf{c})\tilde{\mathbf{y}}}}.$$

From Corollary 4.4, denoting $\hat{m}_{t1}^{(\mathrm{bin})}$ the natural consistent estimate for $m_{t1}^{(\mathrm{bin})}$ (as per Remark 3.6), the optimal class allocation decision for $\mathbf{x}$ reduces to the "averaged-mean" test

$$\mathbf{v}^\mathsf{T}\mathbf{x} = \mathbf{v}(\mathbf{y})^\mathsf{T}\mathbf{x} \underset{\mathcal{C}_{t2}}{\overset{\mathcal{C}_{t1}}{\gtrless}} \frac{1}{2}\left(\hat{m}_{t1}^{(\mathrm{bin})} + \hat{m}_{t2}^{(\mathrm{bin})}\right) \tag{3}$$

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

$$\begin{aligned}\epsilon_t &\equiv P\left(\mathbf{v}^\mathsf{T}\mathbf{x} \underset{\mathcal{C}_{t2}}{\overset{\mathcal{C}_{t1}}{\gtrless}} \frac{1}{2}(\hat{m}_{t1}^{(\mathrm{bin})} + \hat{m}_{t2}^{(\mathrm{bin})})\right) \\ &= \mathcal{Q}\left(\frac{1}{2}(m_{t1}^{(\mathrm{bin})} - m_{t2}^{(\mathrm{bin})})\right) + o(1). \end{aligned} \tag{4}$$

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

$$\begin{aligned}\tilde{\mathbf{y}}_{[t]}^\star &\equiv \underset{\tilde{\mathbf{y}} \in \mathbb{R}^{2k}}{\arg\max} \, (m_{t1}^{(\mathrm{bin})} - m_{t2}^{(\mathrm{bin})})^2 \\ &= D_\mathbf{c}^{-\frac{1}{2}} \left(\mathcal{M} + \mathbf{I}_{2k}\right)^{-1} \mathcal{M} D_\mathbf{c}^{-\frac{1}{2}} (\mathbf{e}_{t1} - \mathbf{e}_{t2}). \end{aligned} \tag{5}$$

Letting $\hat{\tilde{\mathbf{y}}}_{[t]}^\star$ be the natural consistent estimator of $\tilde{\mathbf{y}}_{[t]}^\star$ (again from Remark 3.6), and updating $\mathbf{v} = \mathbf{v}(\tilde{\mathbf{y}}_{[t]})$ accordingly, the corresponding (asymptotically) optimal value $\epsilon_t^\star$ of the error rate ϵ_t is

$$\epsilon_t^\star = \mathcal{Q}\left(\frac{1}{2}\sqrt{(\mathbf{e}_{t1}^{[2k]} - \mathbf{e}_{t2}^{[2k]})^\mathsf{T} \mathcal{H} (\mathbf{e}_{t1}^{[2k]} - \mathbf{e}_{t2}^{[2k]})}\right) + o(1), \tag{6}$$

$$\text{with} \quad \mathcal{H} = D_\mathbf{c}^{-\frac{1}{2}} \mathcal{M} \left(\mathcal{M} + \mathbf{I}_{2k}\right)^{-1} \mathcal{M} D_\mathbf{c}^{-\frac{1}{2}}$$

This formula is instructive to discuss: under strong or weak task correlation, $\tilde{\mathbf{y}}_{[t]}^\star$ implements differing strategies to avoid *negative transfers*. For instance, if $\boldsymbol{\mu}_{tj}^\mathsf{T}\boldsymbol{\mu}_{t'j'} = 0$ 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: $\tilde{\mathbf{y}}_{[t]}^\star$ 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 $\tilde{\mathbf{y}}_{[t]} = [1, -1, \ldots, 1, -1]^\mathsf{T}$ is even more detrimental when $\boldsymbol{\mu}_{tj}^\mathsf{T}\boldsymbol{\mu}_{t'j'} < 0$ for some $t' \neq t'$: when the mapping of classes across tasks is reversed, these tasks work *against* the classification.

Remark 4.5 (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 that, at least under the present data configuration, the proposed SPCA-MTL framework is optimal.

4.3. Binary-based multi-class classification

With 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 $\tilde{\mathbf{y}}_{[t]}^{\star(\ell)}$ as per Equation (5); 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 $\tilde{\mathbf{y}}_{[t]}^{\star(\ell)}$ manages to isolate the value $m_{t\ell}^{(\mathrm{bin})} = m_{t1}^{(\mathrm{bin},\ell)}$ for class $\mathcal{C}_{t\ell} = \mathcal{C}_{t1}^{(\ell)}$ from the values $m_{tj}^{(\mathrm{bin})}$ of all other classes $\mathcal{C}_{tj}$, $j \neq \ell$, to such an extent that (relatively speaking) these $m_{tj}^{(\mathrm{bin})}$ can be considered quite close, and so close to their mean $m_{t2}^{(\mathrm{bin},\ell)}$, without much impact on the classifier performance. Finally, the class allocation for unknown data $\mathbf{x}$ is based on the largest classifier score. But, to avoid biases that naturally arise in the one-versus-all approach (Bishop, 2006, Section 7.1.3), this imposes that the m different classifiers be "comparable and aligned". To this end, we exploit Corollary 4.4 and Remark 3.6, 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 the largest score $\arg\max_\ell \mathbf{v}(\mathbf{y}_{[t]}^{\star(\ell)})^\mathsf{T}\mathbf{x}$ (as conventionally performed (Bishop, 2006, Section 7.1.3)), the class allocation is based on the centered scores $\arg\max_\ell \{V(y_{[t]}^{\star(\ell)})^\mathsf{T}\mathbf{x} - m_{t1}^{(\mathrm{bin},\ell)}\}$.[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 (Lelarge & Miolane, 2019) 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 $\mathbf{X} = [\mathbf{X}_{[1]}, \ldots, \mathbf{X}_{[k]}]$, $\mathbf{X}_{[t']} = [\mathbf{X}_{[t']1}, \ldots, \mathbf{X}_{[t']m}]$, $\mathbf{X}_{[t']\ell} \in \mathbb{R}^{p \times n_{t'\ell}}$ and test $\mathbf{x}$.
Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for target task t.
for $\ell = 1$ **to** m **do**
  Estimate $\mathbf{c}$ and $\mathcal{M}$ (from Remark 3.6) using $\mathbf{X}_{[t']\ell}$ as data of class $\mathcal{C}_{t'1}^{(\ell)}$ for each $t' \in \{1, \ldots, k\}$ and $\{\mathbf{X}_{[t']1}, \ldots, \mathbf{X}_{[t']m}\} \setminus \{\mathbf{X}_{[t']\ell}\}$ as data of class $\mathcal{C}_{t'2}^{(\ell)}$.
  Evaluate labels

$$\tilde{\mathbf{y}}_{[t]}^{\star(\ell)} = D_{\mathbf{c}}^{-\frac{1}{2}} \left(\mathcal{M} + \mathbf{I}_{2k}\right)^{-1} \mathcal{M} D_{\mathbf{c}}^{-\frac{1}{2}} (\mathbf{e}_{t1}^{[2k]} - \mathbf{e}_{t2}^{[2k]}).$$

  Compute the classification score

$$g_{\mathbf{x},t}^{(\ell)} = \tilde{\mathbf{y}}_{[t]}^{\star(\ell)\mathsf{T}} \mathbf{J}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{x} / \| \tilde{\mathbf{y}}_{[t]}^{\star(\ell)\mathsf{T}} \mathbf{J}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \|.$$

  Estimate $m_{t1}^{(\mathrm{bin},\ell)}$ as $\hat{m}_{t1}^{(\mathrm{bin},\ell)}$ from Corollary 4.4.
end for
Output: $\hat{\ell} = \arg\max_{\ell \in \{1,\ldots,m\}} (g_{\mathbf{x},t}^{(\ell)} - \hat{m}_{t1}^{(\mathrm{bin},\ell)})$.

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 (Tiomoko et al., 2020) and CDLS (Hubert Tsai et al., 2016).[7]

Transfer learning for binary classification. First consider a two-task two-class ($k, m = 2$) scenario with $\mathbf{x}_{t\ell}^{(j)} \sim \mathcal{N}((-1)^j \boldsymbol{\mu}_t, \mathbf{I}_p)$, $\boldsymbol{\mu}_2 = \beta \boldsymbol{\mu}_1 + \sqrt{1-\beta^2}\boldsymbol{\mu}_1^\perp$ for $\boldsymbol{\mu}_1^\perp$ any vector orthogonal to $\boldsymbol{\mu}_1$ 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 $\tilde{\mathbf{y}}_{[t]} = [1, -1, \ldots, 1, -1]^{\mathsf{T}}$ (N-SPCA), were added. MTL SPCA properly tracks task relatedness, while CDLS fails when both tasks are similar. MTL LSSVM shows identical performances but at the cost of setting optimal hyperparameters. Probably most importantly, when *not optimizing* the labels $\mathbf{y}$, 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. At the same time, MTL LSSVM and CDLS quickly become prohibitively expensive.

Transfer learning for multi-class classification. We next experiment on the ImageClef dataset (Ionescu et al., 2017) 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 "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 $\boldsymbol{\mu}_{tj} = (-1)^j \boldsymbol{\mu}_{[t]}$ with $\boldsymbol{\mu}_{[t]} = \beta_{[t]}\boldsymbol{\mu} + \sqrt{1-\beta_{[t]}^2}\boldsymbol{\mu}^\perp$ for $\beta_{[t]}$ drawn uniformly at random in $[0, 1]$ and with $\boldsymbol{\mu} = \mathbf{e}_1^{[p]}$, $\boldsymbol{\mu}^\perp = \mathbf{e}_p^{[p]}$. The real-world data are the Amazon review (textual) dataset[8] (Blitzer et al., 2007) and the MNIST (image) dataset (Deng, 2012). 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.

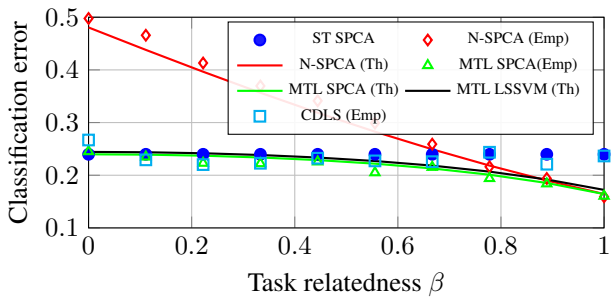
6. Conclusion

Following recent works on large-dimensional statistics for the design of simple, cost-efficient, and tractable machine

[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 (Tan et al., 2018)) are not used for comparison.

[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.



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 $\boldsymbol{\mu}_1 = \mathbf{e}_1^{[p]}$, $\boldsymbol{\mu}_1^\perp = \mathbf{e}_p^{[p]}$, $\boldsymbol{\mu}_2 = \beta\boldsymbol{\mu}_1 + \sqrt{1-\beta^2}\boldsymbol{\mu}_1^\perp$, $p = 100$, $n_{1j} = 1\,000$, $n_{2j} = 50$; **(Right)** running time comparison (in sec); $n = 2p$, $n_{tj}/n = 0.25$. Averaged over 1 000 test samples.

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.

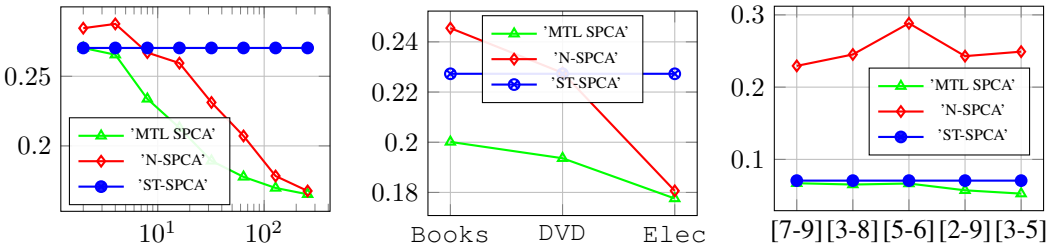
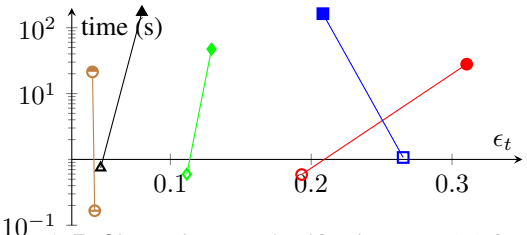


Figure 3. Empirical classification error vs. number of tasks; **(Left)** Synthetic Gaussian with random task correlation: $p = 200$, $n_{11} = n_{12} = 50$, $n_{21} = n_{22} = 5$, 10 000 test samples; **(Center)** Amazon Review: $n_{11} = n_{12} = 100$, $n_{21} = n_{22} = 50$, 2 000 test samples; **(Right)**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



Datasets (Features)	Tasks	Classes	Mark
Synthetic (Gaussian)	3	10	○
Office-Caltech (VGG)	4	10	◇
Office 31 (Resnet-50)	4	31	□
Office-Home (Resnet-50)	3	65	△
Image-Clef (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: $\boldsymbol{\mu}_j = 2\mathbf{e}_j^{[p]}$, $\boldsymbol{\mu}_j^\perp = 2\mathbf{e}_{p-j}^{[p]}$, $\beta_1 = 0.2$, $\beta_2 = 0.4$, $\beta_3 = 0.6$; $p = 200$, $n_{1j} = n_{2j} = 100$, $n_{3j} = 50$; 1 000 test sample averaging.

learning algorithms (Couillet et al., 2021), the article confirms the possibility of achieving 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 framework, practical extensions to semi-supervised learning (when labelled data are scarce) with possibly missing, unbalanced, or incorrectly labelled data are being considered.

References

Ando, R. K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

Ashtiani, H. and Ghodsi, A. A dimension-independent generalization bound for kernel supervised principal component analysis. In *Feature Extraction: Modern Questions and Challenges*, pp. 19–29. PMLR, 2015.

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

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137, 2006.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 769–776, 2013.

Barshan, E., Ghodsi, A., Azimifar, Z., and Jahromi, M. Z. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

Benaych-Georges, F. and Nadakuditi, R. R. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

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

Blitzer, J., Dredze, M., and Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

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

Chao, G., Luo, Y., and Ding, W. Recent advances in supervised dimension reduction: A survey. *Machine learning and knowledge extraction*, 1(1):341–358, 2019.

Couillet, R. and Debbah, M. *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

Couillet, R. and Liao, Z. *Random Matrix Methods for Machine Learning*. Cambridge University Press, 2022.

Couillet, R., Chatelain, F., and Bihan, N. L. Two-way kernel matrix puncturing: towards resource-efficient pca and spectral clustering. *arXiv preprint arXiv:2102.12293*, 2021.

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

El Karoui, N. Random matrices and high-dimensional statistics: beyond covariance matrices. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pp. 2857–2876. World Scientific, 2018.

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

Ghojogh, B. and Crowley, M. Unsupervised and supervised principal component analysis: Tutorial. *arXiv preprint arXiv:1906.03148*, 2019.

Gong, P., Ye, J., and Zhang, C.-s. Multi-stage multi-task feature learning. In *Advances in neural information processing systems*, pp. 1988–1996, 2012.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

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

Hubert Tsai, Y.-H., Yeh, Y.-R., and Frank Wang, Y.-C. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5081–5090, 2016.

Ionescu, B., Müller, H., Villegas, M., Arenas, H., Boato, G., Dang-Nguyen, D.-T., Cid, Y. D., Eickhoff, C., de Herrera, A. G. S., Gurrin, C., et al. Overview of imageclef 2017: Information extraction from images. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 315–337. Springer, 2017.

Johnstone, I. M. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pp. 295–327, 2001.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Lee, S., Zou, F., and Wright, F. A. Convergence and prediction of principal component scores in high-dimensional settings. *Annals of statistics*, 38(6):3605, 2010.

Lelarge, M. and Miolane, L. 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)*, pp. 639–643. IEEE, 2019.

Liu, J., Ji, S., and Ye, J. Multi-task feature learning via efficient l2, 1-norm minimization. *arXiv preprint arXiv:1205.2631*, 2012.

Liu, Q., Liao, X., and Carin, L. Semi-supervised multitask learning. *Advances in Neural Information Processing Systems*, 20:937–944, 2007.

Long, M., Wang, J., Ding, G., Shen, D., and Yang, Q. Transfer learning with graph co-regularization. *IEEE Transactions on Knowledge and Data Engineering*, 26 (7):1805–1818, 2013.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

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

Marchenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

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

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

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

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

Paul, D. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, pp. 1617–1642, 2007.

Rei, M. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*, 2017.

Ritchie, A., Scott, C., Balzano, L., Kessler, D., and Sripada, C. S. Supervised principal component analysis via manifold optimization. In *2019 IEEE Data Science Workshop (DSW)*, pp. 6–10. IEEE, 2019.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pp. 1–4, 2005.

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

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Seddik, M. E. A., Louart, C., Tamaazousti, M., and Couillet, R. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. *arXiv preprint arXiv:2001.08370*, 2020.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Tiomoko, M., Couillet, R., and Tiomoko, H. Large dimensional analysis and improvement of multi task learning. *arXiv preprint arXiv:2009.01591*, 2020.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

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

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

Zhang, W., Li, R., Zeng, T., Sun, Q., Kumar, S., Ye, J., and Ji, S. Deep model based transfer and multi-task learning for biological image analysis. *IEEE transactions on Big Data*, 2016.

Zhang, X., Sun, Q., and Kong, D. Supervised principal component regression for functional response with high dimensional predictors. *arXiv preprint arXiv:2103.11567*, 2021.

Zhang, Y. and Yang, Q. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

Zhang, Y. and Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Zhang, Y. and Yeung, D.-Y. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

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

Abstract

This document contains the main technical arguments omitted in the article's core due to space limitations and is organized as follows. Section A details the large dimensional analysis of PCA. Section B 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 C 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 D. Finally, Section E explains how to use the code to implement the paper's main results.

A. Large dimensional analysis of Single Task PCA

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

Assumption A.1 (Distribution of $\mathbf{X}$ and $\mathbf{x}$)**.** The columns of $\mathbf{X}$ are independent random vectors with $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_m]$, $\mathbf{X}_j = [\mathbf{x}_1^{(j)}, \ldots, \mathbf{x}_{n_j}^{(j)}] \in \mathbb{R}^{p \times n_j}$ where $\mathbf{x}_i^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{I}_p)$. As for $\mathbf{x}$, it follows an independent $\mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \mathbf{I}_p)$ distribution. We will further denote $\mathbf{x} \in \mathcal{C}_j$ to indicate that data vector $\mathbf{x}$ belongs to class j, i.e., $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{I}_p)$.

Assumption A.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 $\mathbf{c} = [c_1, \ldots, c_m]^\mathsf{T}$. Furthermore, the latent feature space dimension τ is constant with respect to n, p.

A.1. Isolated eigenvalues

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

$$\det\left(\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right) = 0.$$

Writing $\mathbf{X} = \mathbf{M}\mathbf{J}^\mathsf{T} + \mathbf{W}$ with $\mathbf{M} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m] \in \mathbb{R}^{p \times m}$, $\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_m]$, where $\mathbf{j}_j = (0, \ldots, 0, \mathbb{1}_{n_j}, 0, \ldots, 0)^\mathsf{T}$ and where $\mathbf{W}$ is a random matrix with independent standard Gaussian entries, this becomes

$$\det\left(\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T} + \mathcal{U}\mathcal{V}^\mathsf{T} - z\mathbf{I}_p\right) = 0, \tag{7}$$

where $\mathcal{U} = \frac{1}{\sqrt{p}}[\mathbf{M}, \mathbf{W}\mathbf{J}] \in \mathbb{R}^{p \times 2m}$ and $\mathcal{V} = \frac{1}{\sqrt{p}}[\mathbf{M}\mathbf{J}^\mathsf{T}\mathbf{J} + \mathbf{W}^\mathsf{T}\mathbf{J}, \mathbf{M}] \in \mathbb{R}^{p \times 2m}$ are low rank matrices (as $n, p \to \infty$); as for $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T}$, its limiting eigenvalue distribution under Assumption 3.2 is known as the Marčenko-Pastur law (Marchenko & Pastur, 1967), recalled next in whole generality:

Theorem A.3. *Let $\mathbf{W}$ 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 $\mu_{\hat{\mathbf{C}}} = \frac{1}{p}\sum_{i=1}^p \delta_{\hat{\lambda}_i}$ *of the eigenvalues $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_p$ of $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T}$, converges weakly, with probability one, to a nonrandom distribution, known as the Marčenko–Pastur law and denoted $\mu_{\text{MP}}^{c_0}$. If $c_0 \in (0, 1)$, $\mu_{\text{MP}}^{c_0}$ has density:*

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

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 μ_{MP}^{1/c_0} with weights $1 - (1/c_0)$ and $1/c_0$.

The spectrum of $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T}$, which contains no structural information (generally referred to as a "noise bulk"), will not be useful for classification. The challenge is to determine which observed eigenvalues represent the class structure. Specifically, let us seek for the presence of an eigenvalue λ_j of $\frac{1}{p}\mathbf{X}\mathbf{X}^\mathsf{T}$ asymptotically greater than the limit $(1 + \sqrt{1/c_0})^2$ of the largest eigenvalue of $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T}$. Following the initial ideas of (Baik & Silverstein, 2006; Benaych-Georges & Nadakuditi, 2012), the

approach is to isolate the low-rank contribution $\mathcal{U}\mathcal{V}^\mathsf{T}$ from the noise matrix $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T}$. Factoring out $\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T} - z\mathbf{I}_p$ and using Sylverster's identity ($\det(\mathbf{AB}+\mathbf{I}) = \det(\mathbf{BA}+\mathbf{I})$), Equation (7) is equivalent to:

$$\det\left(\mathcal{V}^\mathsf{T}\mathbf{Q}(z)\mathcal{U} + \mathbf{I}_{2m}\right) = 0, \quad \text{with} \quad \mathbf{Q}(z) = \left(\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}.$$

We next retrieve the large dimensional limit (or, more specifically, a *deterministic equivalent* (Couillet & Debbah, 2011, Chapter 6)) of $\mathcal{V}^\mathsf{T}\mathbf{Q}(z)\mathcal{U} + \mathbf{I}_{2m}$ under Assumptions A.1 and A.2. Defining the *resolvents* and *co-resolvents* $\mathbf{Q}(z) = (\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T} - z\mathbf{I}_p)^{-1}$ and $\tilde{\mathbf{Q}}(z) = (\frac{1}{p}\mathbf{W}^\mathsf{T}\mathbf{W} - z\mathbf{I}_n)^{-1}$, as $n, p \to \infty$ with $p/n \to c \in (0, \infty)$, we have

$$\mathbf{Q}(z) \leftrightarrow \bar{\mathbf{Q}}(z),\ \bar{\mathbf{Q}}(z) = \delta(z)\mathbf{I}_p$$
$$\tilde{\mathbf{Q}}(z) \leftrightarrow \bar{\tilde{\mathbf{Q}}}(z),\ \bar{\tilde{\mathbf{Q}}}(z) = \tilde{\delta}(z)\mathbf{I}_n$$

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

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

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

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

$$\begin{aligned}&\det\left(\mathcal{V}^\mathsf{T}\mathbf{Q}(z)\mathcal{U} + \mathbf{I}_{2m}\right)\\ &= \det\begin{pmatrix}\mathbf{I}_m + \frac{1}{p}\mathbf{J}^\mathsf{T}\mathbf{J}\mathbf{M}^\mathsf{T}\mathbf{Q}(z)\mathbf{M} + \frac{1}{p}\mathbf{J}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{Q}(z)\mathbf{M} & \frac{1}{p}\mathbf{J}^\mathsf{T}\mathbf{J}\mathbf{M}^\mathsf{T}\mathbf{Q}(z)\mathbf{W}\mathbf{J} + \frac{1}{p}\mathbf{J}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{Q}(z)\mathbf{W}\mathbf{J}\\ \frac{1}{p}\mathbf{M}^\mathsf{T}\mathbf{Q}(z)\mathbf{M} & \mathbf{I}_m + \frac{1}{p}\mathbf{M}^\mathsf{T}\mathbf{Q}(z)\mathbf{W}\mathbf{J}\end{pmatrix}\end{aligned}$$

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

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

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

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

where $\mathcal{M} = \lim_{p\to\infty} \frac{1}{c_0} D_{\mathbf{c}}^{\frac{1}{2}}\mathbf{M}^\mathsf{T}\mathbf{M}D_{\mathbf{c}}^{\frac{1}{2}}$. 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.

A.2. PCA projectors

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

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

$$\boldsymbol{\mu}_j^\mathsf{T}\mathbf{u}_i = \frac{\boldsymbol{\mu}_j^\mathsf{T}\mathbf{u}_i\mathbf{u}_i^\mathsf{T}\mathbb{1}_p/p}{\sqrt{\mathbb{1}_p^\mathsf{T}\mathbf{u}_i\mathbf{u}_i^\mathsf{T}\mathbb{1}_p/p^2}} \tag{8}$$

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

$$\begin{aligned}
\mathbf{a}^\mathsf{T}\mathbf{u}_i\mathbf{u}_i^\mathsf{T}\frac{\mathbb{1}_p}{p} &= \frac{-1}{2\pi\imath}\oint_{\gamma_i} \mathbf{a}^\mathsf{T}\left(\frac{1}{p}\mathbf{W}\mathbf{W}^\mathsf{T} + \mathcal{U}\mathcal{V}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\frac{\mathbb{1}_p}{p} \\
&= \frac{-1}{2\pi\imath}\oint_{\gamma_i} \mathbf{a}^\mathsf{T}\left(\mathbf{Q}(z) - \mathbf{Q}(z)\mathcal{U}\left(\mathbf{I}_{2m} + \mathcal{V}^\mathsf{T}\mathbf{Q}(z)\mathcal{U}\right)^{-1}\mathcal{V}^\mathsf{T}\mathbf{Q}(z)\right)\frac{\mathbb{1}_p}{p} \\
&= \frac{1}{2\pi\imath}\oint_{\gamma_i} \mathbf{a}^\mathsf{T}\mathbf{Q}(z)\mathcal{U}\left(\mathbf{I}_{2m} + \mathcal{V}^\mathsf{T}\mathbf{Q}(z)\mathcal{U}\right)^{-1}\mathcal{V}^\mathsf{T}\mathbf{Q}(z)\frac{\mathbb{1}_p}{p}
\end{aligned}$$

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

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

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

Altogether, this gives :

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

with $\bar{\mathbf{u}}_i$ 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

$$\mathbf{a}^\mathsf{T}\mathbf{u}_i\mathbf{u}_i^\mathsf{T}\frac{\mathbb{1}_p}{p} \leftrightarrow \frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)}\mathbf{a}^\mathsf{T}\mathbf{M}D_{\mathbf{c}}^{\frac{1}{2}}\bar{\mathbf{u}}_i\bar{\mathbf{u}}_i^\mathsf{T}D_{\mathbf{c}}^{\frac{1}{2}}\mathbf{M}\frac{\mathbb{1}_p}{p}.$$

Finally, using Equation (8), we conclude

$$\boldsymbol{\mu}_j^\mathsf{T}\mathbf{u}_i \xrightarrow{\text{a.s.}} \sqrt{\frac{c_0\ell_i - 1}{\ell_i^2(\ell_i+1)}}\bar{\mathbf{u}}_i^\mathsf{T}\mathcal{M}D_{\mathbf{c}}^{-\frac{1}{2}}\mathbf{e}_j^{[m]}.$$

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

B. Large dimensional analysis of Multi-Task SPCA

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

Assumption B.1 (Distribution of $\mathbf{X}$)**.** For class j of task t, denoted $\mathcal{C}_{tj}$, $\mathbf{x}_{t\ell}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}_{tj}, \boldsymbol{\Sigma}_{tj})$, for some $\boldsymbol{\mu}_{tj} \in \mathbb{R}^p$. We further denote $\mathbf{M} \equiv [\boldsymbol{\mu}_{11}, \ldots, \boldsymbol{\mu}_{km}] \in \mathbb{R}^{p\times mk}$.

Assumption B.2 (Growth Rate)**.** As $n \to \infty$, $p/n \to c_0 > 0$ and, for $1 \leq j \leq m$, $n_{tj}/n \to c_{tj} > 0$; we denote $\mathbf{c} = [c_{11}, \ldots, c_{km}]^\mathsf{T} \in \mathbb{R}^{km}$, and $D_\mathbf{c} = \mathrm{diag}(\mathbf{c})$. Besides,

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

B.1. Isolated eigenvalues

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

$$\det\left(\frac{1}{p} \mathbf{X}\mathbf{J} \frac{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T}}{n} \mathbf{J}^\mathsf{T} \mathbf{X}^\mathsf{T} - z\mathbf{I}_p \right) = \det\left(\frac{1}{p} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \mathbf{J}^\mathsf{T} \frac{\mathbf{X}^\mathsf{T}\mathbf{X}}{n} \mathbf{J} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} - z\mathbf{I}_m \right)$$

Besides, we have

$$\frac{1}{n} \mathbf{J}^\mathsf{T} \frac{\mathbf{X}^\mathsf{T}\mathbf{X}}{p} \mathbf{J} \leftrightarrow \frac{1}{n} \mathbf{J}^\mathsf{T} D_{\tilde{\mathbf{v}}} \mathbf{J} + \frac{1}{c_0} D_\mathbf{c} \mathbf{M}^\mathsf{T} \mathbf{M} D_\mathbf{c}$$

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

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

$$\mathcal{H} = (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \left(D_\mathbf{c} + D_\mathbf{c}^{\frac{1}{2}} \mathcal{M} D_\mathbf{c}^{\frac{1}{2}} \right) (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}.$$

B.2. SPCA projectors

Computation of the mean. Since the eigenvector $\mathbf{v}_i$ is defined up to sign, we may, as above, impose that

$$\boldsymbol{\mu}_{tj}^\mathsf{T} \mathbf{v}_i = \frac{\boldsymbol{\mu}_{tj}^\mathsf{T} \mathbf{v}_i \mathbf{v}_i^\mathsf{T} \mathbb{1}_p / p}{\sqrt{\mathbb{1}_p^\mathsf{T} \mathbf{v}_i \mathbf{v}_i^\mathsf{T} \mathbb{1}_p / p^2}}. \tag{9}$$

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

$$\begin{aligned}
\mathbf{a}^\mathsf{T} \mathbf{v}_i \mathbf{v}_i^\mathsf{T} \frac{\mathbb{1}_p}{p} &= \frac{-1}{2\pi\imath} \oint_{\gamma_i} \mathbf{a}^\mathsf{T} \left(\frac{1}{p} \mathbf{X}\mathbf{J} \frac{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T}}{n} \mathbf{J}^\mathsf{T} \mathbf{X}^\mathsf{T} - z\mathbf{I}_p \right)^{-1} \frac{\mathbb{1}_p}{p} \\
&= \frac{1}{2\pi\imath} \oint_{\gamma_i} \frac{1}{z} \mathbf{a}^\mathsf{T} \frac{1}{np} \mathbf{X}\mathbf{J} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \left(z\mathbf{I}_m - \frac{1}{np} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \mathbf{J}^\mathsf{T} \mathbf{X}^\mathsf{T} \mathbf{X} \mathbf{J} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \right)^{-1} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \mathbf{J}^\mathsf{T} \mathbf{X}^\mathsf{T} \frac{\mathbb{1}_p}{p} \\
&= \frac{1}{2\pi\imath c_0} \oint_{\gamma_i} \frac{1}{z} \mathbf{a}^\mathsf{T} \mathbf{M} D_\mathbf{c} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \left(z\mathbf{I}_m - \mathcal{H} \right)^{-1} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} D_\mathbf{c} \mathbf{M}^\mathsf{T} \frac{\mathbb{1}_p}{p} + o(1) \\
&= \frac{1}{c_0} \frac{1}{\bar{\lambda}_i} \mathbf{a}^\mathsf{T} \mathbf{M} D_\mathbf{c} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} \bar{\mathbf{v}}_i \bar{\mathbf{v}}_i^\mathsf{T} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} D_\mathbf{c} \mathbf{M}^\mathsf{T} \frac{\mathbb{1}_p}{p} + o(1)
\end{aligned}$$

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

Therefore,

$$\boldsymbol{\mu}_{tj}^\mathsf{T} \mathbf{v}_i \xrightarrow{\text{a.s.}} \sqrt{\frac{1}{\bar{\lambda}_i}} \bar{\mathbf{v}}_i^\mathsf{T} (\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} D_\mathbf{c}^{\frac{1}{2}} \mathcal{M} D_\mathbf{c}^{-\frac{1}{2}} \mathbf{e}_{tj}^{[mk]}.$$

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

$$\begin{aligned}
\mathbf{v}_i^\mathsf{T}\boldsymbol{\Sigma}_{tj}\mathbf{v}_i &= \mathrm{tr}\left(\mathbf{v}_i\mathbf{v}_i^\mathsf{T}\boldsymbol{\Sigma}_{tj}\right) \\
&= \frac{-1}{2\pi\imath}\mathrm{tr}\left(\boldsymbol{\Sigma}_{tj}\oint_{\gamma_i}\left(\frac{1}{p}\mathbf{X}\mathbf{J}\frac{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T}}{n}\mathbf{J}^\mathsf{T}\mathbf{X}^\mathsf{T} - z\mathbf{I}_p\right)^{-1}\right) \\
&= \frac{1}{2\pi\imath}\mathrm{tr}\left(\boldsymbol{\Sigma}_{tj}\oint_{\gamma_i}\frac{1}{npz}\mathbf{X}\mathbf{J}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\left(z\mathbf{I}_m - \frac{1}{np}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\mathbf{J}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{J}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\right)^{-1}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\mathbf{J}^\mathsf{T}\mathbf{X}^\mathsf{T}\right) \\
&= \frac{1}{2\pi\imath}\mathrm{tr}\left(\oint_{\gamma_i}\frac{1}{npz}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\mathbf{J}^\mathsf{T}\mathbf{X}^\mathsf{T}\boldsymbol{\Sigma}_{tj}\mathbf{X}\mathbf{J}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\left(z\mathbf{I}_m - \frac{1}{np}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\mathbf{J}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{J}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\right)^{-1}\right) \\
&= \frac{1}{\bar{\lambda}_i}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}}\mathcal{T}_{tj}(\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\mathsf{T})^{\frac{1}{2}} + o(1)
\end{aligned}$$

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

When $\boldsymbol{\Sigma}_{tj} = \mathbf{I}_p$, as treated in the main article, it is immediate that $\mathrm{Var}(\mathbf{u}_i^\mathsf{T}\mathbf{x}) = 1$.

C. 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.

C.1. One-versus-all multi-class preliminary

The literature (Bishop, 2006) 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 (Bishop, 2006, 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 5 for a convenient illustration of the improvement induced by this centering-scaling and label optimization approach.

C.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 5, 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 5. 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 5: the result of the center-scale operation appears in the second column of Figure 5.

This first improvement step simplifies the algorithm which now boils down to determining the index of the largest $g_{\mathbf{x},t}^{(\ell)} - m_{t1}^{(bin,\ell)}$, $\ell \in \{1,\dots,m\}$, 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 $\mathbf{y} = \mathbf{J}\tilde{\mathbf{y}}$, with now $\tilde{\mathbf{y}} \in \mathbb{R}^{mk}$ (instead of $\mathbb{R}^{2k}$) for $\mathbf{J} = \begin{pmatrix} \mathbb{1}_{n_{11}} & & \\ & \ddots & \\ & & \mathbb{1}_{n_{mk}} \end{pmatrix}$;
2. the procedure demands to simultaneously adapt all input scores $\tilde{\mathbf{y}}_{[t]}^{(1)},\dots,\tilde{\mathbf{y}}_{[t]}^{(m)}$.

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

Corollary C.1 (One-versus-all Binary MTL Supervised Principal Component Analysis)**.** *Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{tj}, \mathbf{I}_p)$ independent of $\mathbf{X}$. Then, under Assumptions B.1-4.2 and the above setting, as $p, n \to \infty$,*

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

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

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 5, where the distances between the rightmost Gaussians and the other two are 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 $\ell \in \{1,\dots,m\}$ the optimization problem:

$$\tilde{\mathbf{y}}_{[t]}^{\star(\ell)} = \max_{\tilde{\mathbf{y}}_{[t]}^{(\ell)} \in \mathbb{R}^{km}} \min_{j \neq \ell} \left(m_{t\ell}^{(\mathrm{bin}),\ell} - m_{tj}^{(\mathrm{bin}),\ell} \right) \tag{10}$$

Being a non-convex and non-differentiable (due to the max) optimization, Equation (10) cannot be solved straightforwardly. An approximated solution consists in relaxing the max operator $\max(x_1,\dots,x_n)$ 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 $i' \in \{1,\dots,m\}$, 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 $\tilde{\mathbf{y}}_{[t]}^{(\ell)} \in \mathbb{R}^{2k}$ by "merging" all Gaussians of class $\mathcal{C}_{tj}$ with $j \neq \ell$ into a unique *approximated* Gaussian class with mean $\sum_{j\neq\ell} \frac{n_{tj}}{n-n_{t\ell}} \boldsymbol{\mu}_{tj}$. We may then (abusively) apply Corollary 1, leading to an explicit expression of the optimal label $\tilde{\mathbf{y}}_{[t]}^{\star(\ell)}$, from which Algorithm 1 in the main article unfolds.

Figure 6 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 $\boldsymbol{\mu}_{1j} = \boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_{2j} = \beta\boldsymbol{\mu}_j + \sqrt{1-\beta^2}\boldsymbol{\mu}_j^{\perp}$.

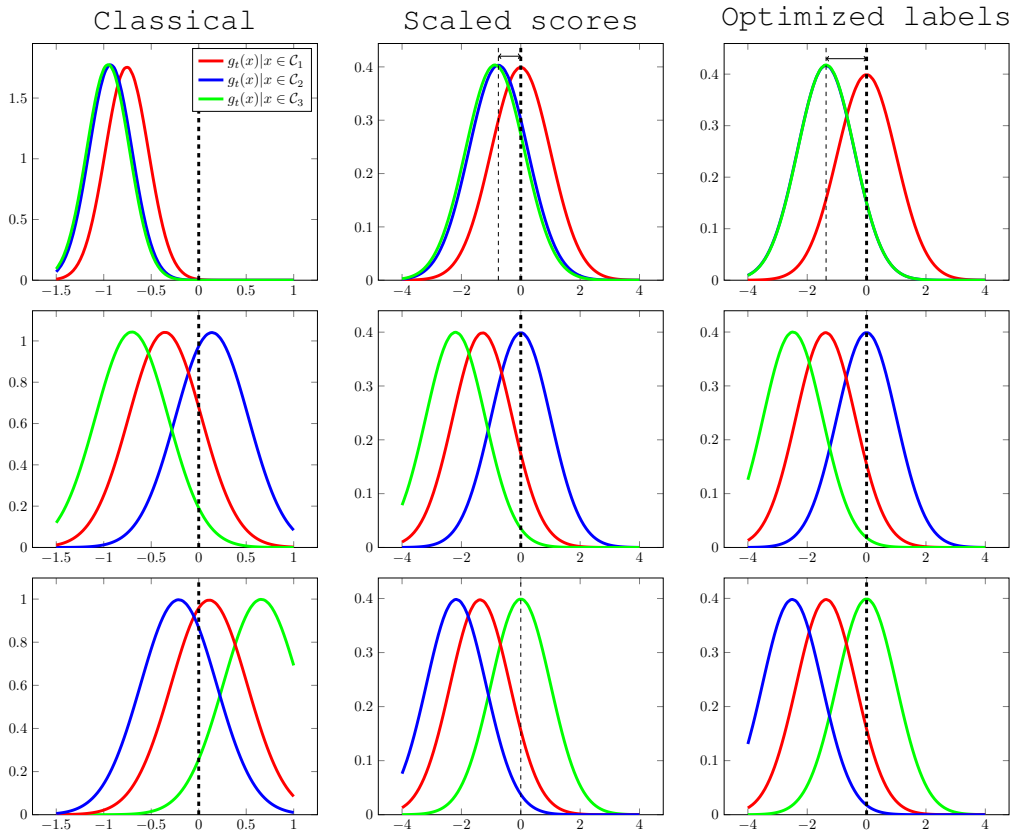


Figure 5. Test score distribution in a 2-task and 3 classes-per-task setting, using a one-versus-all multi-class classification. Every graph in row ℓ depicts the limiting distributions of $g^{(\ell)}_{\mathbf{x},t}$ for $\mathbf{x}$ in different classes. Column 1 (Classical) is the standard implementation 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) $\tilde{\mathbf{y}}^{\star(\ell)}_{[t]}$. Under the "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 the blue curve in 1st row; and similarly for the green curve in 3rd versus 1st row).

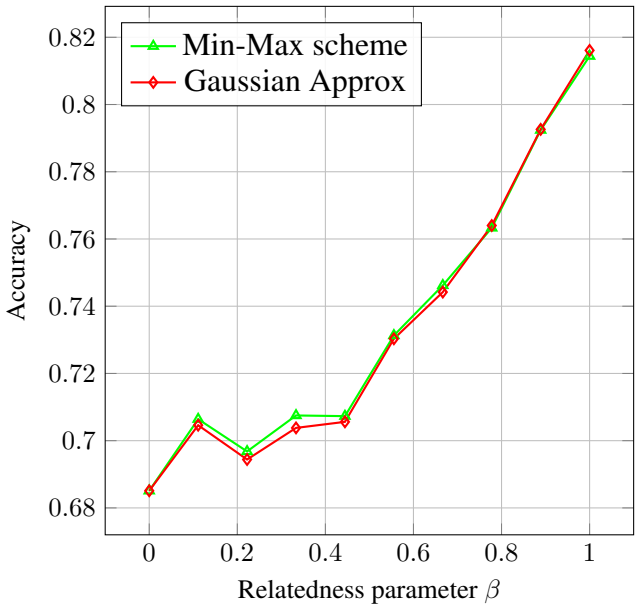


Figure 6. Empirical accuracy as function of the relatedness parameter β on Synthetic Gaussian with $p = 500$, $\boldsymbol{\mu}_j = 3\mathbf{e}_j^{[p]}$, $\boldsymbol{\mu}_j^{\perp} = 3\mathbf{e}_{p-j}^{[p]}$, $n_{1j} = 100$, $n_{2j} = 50$ for $1 \leq j \leq 10$; 10 000 test sample averaging

D. Supplementary experiments

We next experiment on two transfer learning datasets:

- the Office31 dataset (Saenko et al., 2010) 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 (Venkateswara et al., 2017) 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 2 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.

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 2. 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.

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

E. Readme Code

This document explains how to use the code implementing the Random Matrix Improved Multi-Task Learning Supervised Principal Component Analysis (RMT-MTLSPCA) proposed in the core of the article.

E.1. Archive content

- The function implementing the binary version of our method is called `RMTMTLSPCA_binary_train.m` which trains the MTL SPCA proposed algorithm.
- The function implementing our method is called `RMTMTLSPCA_multiclass_train.m` which trains the MTL SPCA proposed algorithm in the multi-class classification.
- The main script comparing the performance of SPCA and PCA in a single task setting `PCA_versus_SPCA_single_task`.
- The main script comparing all algorithms for synthetic data for binary transfer learning `compareTL_binary`.
- The main script comparing all algorithms for synthetic data/real data for multi-class transfer learning is `CompareTL_multiclass.m`.
- The main script illustrating the benefit of adding more tasks to the MTL framework `Increased_tasks.m`.
- The main script illustrating the tradeoff performance versus running time in the multi-task multi-class classification `TradeOff_performance_running_time.m`
- Folder `utils`: containing alternative MTL algorithms among which MMDT algorithm from (Hoffman et al., 2013), and MTL-LSSVM algorithm(Tiomoko et al., 2020)[10] and other functions used for the proposed method.
- Folder `datasets`: containing Office+Caltech dataset, OfficeHome, ImageClef, Office31 and Amazon review dataset. Due to space limitations, the dataset is not included but can be downloaded and put in the corresponding folder. Codes and datasets used are publicly accessible and under free licenses.

E.2. Reproducing the results of the article

The following sections detail the parameter set to reproduce the experiments of the main article.

E.2.1. Figure 1

Script → `PCA_versus_SPCA_single_task.m`

E.3. Figure 2

Script → `compareTL_binary.m`

[10]To use these codes, one needs to have a Matlab compiler for Mex files

E.3.1. TABLE 1

Script → `CompareTL_multiclass.m`
number_trials → 20
dataset → 'synthetic' (for synthetic data)
dataset → 'officehome' (for OfficeHome dataset)
dataset → 'ImageClef' (for ImageClef dataset)

E.3.2. FIGURE 3

Script → `Increased_tasks.m`
dataset → 'synthetic' (for synthetic data)
dataset → 'nlp' (for amazon-review dataset)
dataset → 'mnist' (for MNIST dataset)

E.3.3. FIGURE 4

Script → `TradeOff_performance_running_time.m`
dataset → 'synthetic' (for synthetic data)
dataset → 'officehome' (for OfficeHome dataset)
dataset → 'ImageClef' (for ImageClef dataset)