PCA-based Multi Task Learning: a Random Matrix Approach

Malik Tiomoko, Romain Couillet, Frédéric Pascal

PCA-based Multi Task Learning: a Random Matrix Approach

Malik Tiomoko
Université Paris-Saclay
CentraleSupélec, L2S
91190, Gif-sur-Yvette, France.
malik.tiomoko@u-psud.fr

Romain Couillet
Gipsa Lab
Université Grenoble Alpes

Frédéric Pascal
Université Paris-Saclay
CentraleSupélec, L2S
91190, Gif-sur-Yvette, France

Abstract

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

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

1 Introduction

From single to multiple task learning. Advanced supervised machine learning algorithms require large amounts of *labelled* samples to achieve high accuracy, which in practice is often too demanding. Multi-task learning (MTL) [12, 54, 55] and *transfer learning* provide a potent workaround by appending extra *somewhat similar* datasets to the scarce available dataset of interest. The additional data possibly being of a different nature, MTL effectively solves multiple tasks *in parallel* while exploiting task relatedness to enforce collaborative learning.

State-of-the-art of MTL. To proceed, MTL solves multiple related tasks and introduces shared hyperparameters or feature spaces, optimized to improve the performance of the individual tasks. The crux of efficient MTL lies in both enforcing and, most importantly, evaluating task relatedness: this, in general, is highly non-trivial as this implies to theoretically identify the common features of the datasets. Several heuristics have been proposed which may be split into two groups: parameter- versus feature-based MTL. In parameter-based MTL, the tasks are assumed to share common hyperparameters [16, 51] (*e.g.*, separating hyperplanes in a support vector machine (SVM) flavor) or hyperparameters derived from a common prior distribution [56, 57]. Classical learning mechanisms (SVM, logistic regression, etc.) can be appropriately turned into an MTL version by enforcing parameter relatedness: [16, 51, 37] respectively adapt the SVM, least square-SVM (LS-SVM), and large margin nearest neighbor (LMNN) methods into an MTL paradigm. In feature-based MTL, the data are instead assumed to share a common low-dimensional representation, which needs be identified: through sparse coding, deep neural network embeddings, principal component analysis (PCA) [2, 34, 52, 36] or simply by feature selection [35, 50, 19].

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 [42, 31]; this often occurs when task relatedness is weaker than assumed and MTL enforces fictitious similarities.

A large dimensional analysis to improve MTL. Based on a large dimensional random matrix setting, this work focuses on an elementary (yet powerful) PCA-based MTL approach and provides an exact (asymp-

totic) evaluation of its performance. This analysis conveys insights into the MTL inner workings, which in turn provides an optimal data labelling scheme to fully avert negative transfer.

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

This finding goes in the direction of the compellingly needed development of cost-efficient and environment-friendly AI solutions [26, 46, 21].

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

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

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

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

2 Related works

A series of supervised (single-task) learning methods were proposed which rely on PCA [7, 41, 53, 17]: the central idea is to project the available data onto a shared low-dimensional space, thus ignoring individual data variations. These algorithms are generically coined supervised principal component analysis (SPCA). Their performances are however difficult to grasp as they require to understand the statistics of the PCA eigenvectors: only recently have large dimensional statistics, and specifically random matrix theory, provided first insights into the behavior of eigenvalues and eigenvectors of sample covariance and kernel matrices [8, 25, 4, 27, 39]. To the best of our knowledge, none of these works have drawn an analysis of SPCA: the closest work is likely [3] which however only provides statistical bounds on performance rather than exact results.

On the MTL side, several methods were proposed under unsupervised [32, 45, 6], semi-supervised [40, 30] and supervised (parameter-based [48, 16, 51, 1] or feature-based [2, 29]) flavors. Although most of these works generally achieve satisfying performances on both synthetic and real data, few theoretical analyses and guarantees exist, so that instances of negative transfer are likely to occur.

To be exhaustive, we must mention that, for specific types of data (images, text, time-series) and under the availability of numerous labelled samples, deep learning MTL methods have recently been devised [43]. These are however at odds with the article requirement to leverage scarce labelled samples and to be valid for generic inputs (beyond images or texts): these methods cannot be compared on even grounds with the methods discussed in the present study.[1]

3 Supervised principal component analysis: single task preliminaries

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

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

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

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

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

To predict the label $\mathbf{y}$ of a test data vector $\mathbf{x}$, a simple method to exploit PCA consists in projecting $\mathbf{x}$ onto the PCA subspace U and in performing classification in the projected space. This has the strong advantage to provide a (possibly dramatic) dimensionality reduction (from p to τ) to supervised learning mechanisms, thus improving cost efficiency while mitigating the loss incurred by the reduction in dimension. Yet, the PCA step is fully unsupervised and does not exploit the available class information. It is instead proposed in [7, 13] to trade U for a more representative projector V which "maximizes the dependence" between the projected data $V^\mathsf{T}X$ and the output labels $y = [y_1, \ldots, y_n]^\mathsf{T} \in \mathbb{R}^{n\times m}$. To this end, [7] exploits the Hilbert-Schmidt independence criterion [20], with corresponding optimization

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

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

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

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

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

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

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

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

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

Proposition 1 (Eigenvalue Phase transition)**.** *Under Assumptions 1-2, as $n, p \to \infty$, the* empirical spectral measure $\frac{1}{p}\sum_{i=1}^p \delta_{\lambda_i}$ *of the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_p$ of $\frac{XX^\mathsf{T}}{p}$ converges weakly, with probability one, to the Marčenko-Pastur law [33] 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]

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

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

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

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

$$U^\mathsf{T}\mathbf{x} - G_j \to 0, \quad G_j \sim \mathcal{N}(\mathfrak{m}_j^{(\text{pca})}, I_\tau), \quad \text{in probability,}$$

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

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

As such, only the projections on the eigenvectors of $\frac{1}{p}XX^\mathsf{T}$ attached to *isolated* eigenvalues carry informative discriminating features. Practically, for all n, p

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

large, it is thus useless to perform PCA on a larger dimension than the number of isolated eigenvalues, i.e., $\tau \leq \arg\max_{1\leq i\leq m}\{\ell_i \geq 1/\sqrt{c_0}\}$.

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

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

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

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

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

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

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

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

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

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

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

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

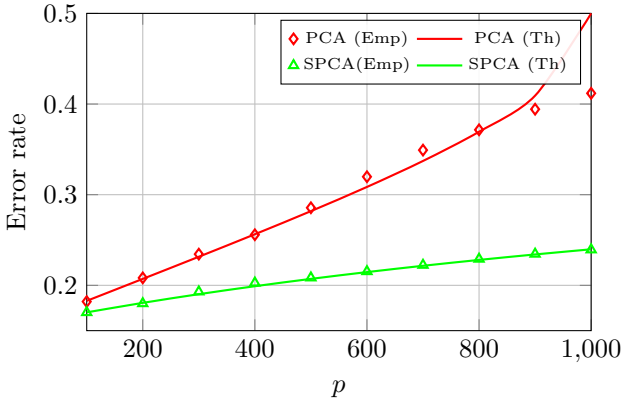


Figure 1: Theoretical (Th) vs. empirical (Emp) error for PCA- and SPCA-based binary classification: $x_i^{(\ell)} \sim \mathcal{N}((-1)^\ell\mu, I_p)$ ($\ell \in \{1, 2\}$), $\mu = e_1^{[p]}$, $n_1 = n_2 = 500$. Averaged over 1 000 test samples.

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

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

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

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

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

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

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

4.1 Multi-class setting

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

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

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

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

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

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

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

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

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

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

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

4.2 Binary classification and optimal labels

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

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

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

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

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

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

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

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

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

$$\begin{aligned}\tilde{y}_{[t]}^\star &\equiv \arg\max_{\tilde{y}\in\mathbb{R}^{2k}} (\mathfrak{m}_{t1}^{(\text{bin})} - \mathfrak{m}_{t2}^{(\text{bin})})^2 \\ &= \mathcal{D}_c^{-\frac{1}{2}} (\mathcal{M} + I_{2k})^{-1} \mathcal{M}\mathcal{D}_c^{-\frac{1}{2}} (e_{t1} - e_{t2}). \end{aligned} \quad (5)$$

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

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

$$\text{with} \quad \mathcal{H} = \mathcal{D}_c^{-\frac{1}{2}} \mathcal{M} (\mathcal{M} + I_{2k})^{-1} \mathcal{M}\mathcal{D}_c^{-\frac{1}{2}}$$

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

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

4.3 Binary-based multi-class classification

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

4.4 Complexity of the SPCA-MTL algorithm

Algorithm 1 is simple to implement and, with optimal hyperparameters consistently estimated, does not require learning by cross validation. The algorithm computational cost is thus mostly related to the computation of the decision scores $g_{\boldsymbol{x},t}^{(\ell)}$, i.e., to a matrix-vector multiplication with matrix size $p \times n$ of complexity $\mathcal{O}(n^2)$ (recall that $p \sim n$). This is quite unlike competing methods: MTL-LSSVM proposed in [48] solves a system of n linear equations, for a complexity of order $\mathcal{O}(n^3)$; MTL schemes derived from SVM (CDLS [23], MMDT [22]) also have a similar $O(n^3)$ complexity, these algorithms solving a quadratic programming problem [11]; besides, in these works, a step of model selection via cross validation needs be performed, which increases the algorithm complexity while simultaneously discarding part of the training data for validation.

[5] The result builds on recent advances in physics-inspired (spin glass models) large dimensional statistics; see for instance [28] for a similar result in a single task semi-supervised learning setting. Being a parallel work of the same authors, the reference is concealed in the present version to maintain anonymity.

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

Algorithm 1 Proposed multi-class MTL SPCA algorithm.

Input: Training $X = [X_{[1]}, \ldots, X_{[k]}]$, $X_{[t']} = [X_{[t']1}, \ldots, X_{[t']m}]$, $X_{[t']\ell} \in \mathbb{R}^{p \times n_{t'\ell}}$ and test $\mathbf{x}$.

Output: Estimated class $\hat{\ell} \in \{1, \ldots, m\}$ of $\mathbf{x}$ for target Task t.

Center and normalize the data per task using z-score normalization [38].

for $\ell = 1$ **to** m **do**

Estimate c and $\mathcal{M}$ (from Remark 1) using $X_{[t']\ell}$ as data of class $\mathcal{C}_{t'1}^{(\ell)}$ for each $t' \in \{1, \ldots, k\}$ and $\{X_{[t']1}, \ldots, X_{[t']m}\} \setminus \{X_{[t']\ell}\}$ as data of class $\mathcal{C}_{t'2}^{(\ell)}$.

Evaluate labels $\tilde{y}_{[t]}^{\star(\ell)} = \mathcal{D}_c^{-\frac{1}{2}} (\mathcal{M} + I_{2k})^{-1} \mathcal{M} \mathcal{D}_c^{-\frac{1}{2}} (e_{t1}^{[2k]} - e_{t2}^{[2k]})$.

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

Estimate $\mathfrak{m}_{t1}^{(\mathrm{bin},\ell)}$ as $\hat{\mathfrak{m}}_{t1}^{(\mathrm{bin},\ell)}$ from Corollary 1.

end for

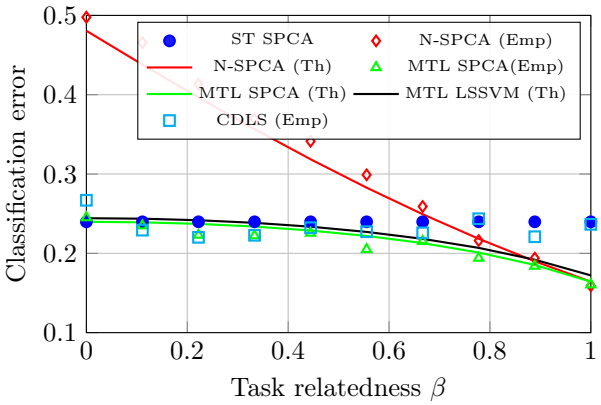
Output: $\hat{\ell} = \arg\max_{\ell \in \{1,\ldots,m\}} (g_{\boldsymbol{x},t}^{(\ell)} - \hat{\mathfrak{m}}_{t1}^{(\mathrm{bin},\ell)})$.

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 [48] and CDLS [23].[7]

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

[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 [47]) are not 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: **(Top)** Theoretical (Th)/empirical (Emp) error rate for 2-class Gaussian mixture transfer with means $\mu_1 = e_1^{[p]}$, $\mu_1^\perp = e_p^{[p]}$, $\mu_2 = \beta\mu_1 + \sqrt{1-\beta^2}\mu_1^\perp$, $p = 100$, $n_{1j} = 1\,000$, $n_{2j} = 50$; **(Bottom)** running time comparison (in sec); $n = 2p$, $n_{tj}/n = 0.25$. Averaged over 1 000 test samples.

Transfer learning for multi-class classification. We next experiment on the ImageClef dataset [24] 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 $\mu_{tj} = (-1)^j \mu_{[t]}$ with $\mu_{[t]} = \beta_{[t]}\mu + \sqrt{1-\beta_{[t]}^2}\mu^\perp$ for $\beta_{[t]}$ drawn uniformly at random in $[0, 1]$ and with $\mu = e_1^{[p]}$, $\mu^\perp = e_p^{[p]}$.

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.

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*

The real-world data are the Amazon review (textual) dataset[8] [10] and the MNIST (image) dataset [15]. 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.

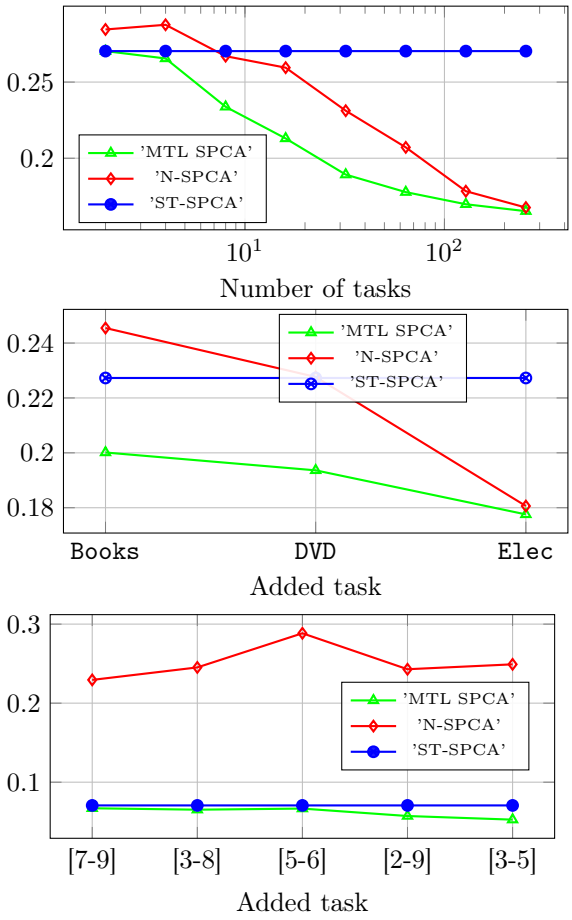


Figure 3: Empirical classification error vs. number of tasks; **(Top)** Synthetic Gaussian with random task correlation: $p = 200$, $n_{11} = n_{12} = 50$, $n_{21} = n_{22} = 5$, 10 000 test samples; **(Center)** Amazon Review: $n_{11} = n_{12} = 100, n_{21} = n_{22} = 50$, 2 000 test samples; **(Bottom)**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

6 Conclusion

Following recent works on large dimensional statistics for the design of simple, cost-efficient, and tractable machine learning algorithms [14], the article confirms the possibility to achieve high performance levels while theoretically averting the main sources of biases, here for the a priori difficult concept of multi-task learning. The article, we hope, will be followed by further investigations of sustainable AI algorithms, driven by modern mathematical tools. In the present multi-task learning framework, practically realistic extensions to semi-supervised learning (when labelled data are scarce) with possibly missing, unbalanced, or incorrectly labelled data are being considered by the authors.

References

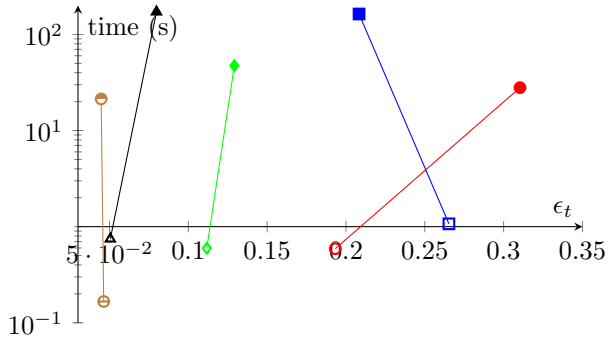
[1] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

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

[3] Hassan Ashtiani and Ali Ghodsi. A dimension-

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

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



Datasets (Features)	Tasks	Classes	Mark
Synthetic (Gaussian)	3	10	○
Office-Caltech[18] (VGG)	4	10	◇
Office 31[44] (Resnet-50)	4	31	□
Office-Home[49] (Resnet-50)	3	65	△
Image-Clef[24] (Resnet-50)	3	12	⊖

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

independent generalization bound for kernel supervised principal component analysis. In *Feature Extraction: Modern Questions and Challenges*, pages 19–29. PMLR, 2015.

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

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

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

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

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

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

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

[11] Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, 3(1):301–320, 2007.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[51] Shuo Xu, Xin An, Xiaodong Qiao, Lijun Zhu, and Lin Li. Multi-output least-squares support vector regression machines. *Pattern Recognition Letters*, 34:1078–1084, 07 2013.

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

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

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

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

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

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