Wasserstein regularization for sparse multi-task regression

Hicham Janati, Marco Cuturi, Alexandre Gramfort

Wasserstein regularization for sparse multi-task regression

Hicham Janati
Inria

Marco Cuturi
Google and CREST / ENSAE

Alexandre Gramfort
Inria

Abstract

We focus in this paper on high-dimensional regression problems where each regressor can be associated to a location in a physical space, or more generally a generic geometric space. Such problems often employ sparse priors, which promote models using a small subset of regressors. To increase statistical power, the so-called multi-task techniques were proposed, which consist in the simultaneous estimation of several related models. Combined with sparsity assumptions, it lead to models enforcing the active regressors to be shared across models, thanks to, for instance ℓ_1/ℓ_q norms. We argue in this paper that these techniques fail to leverage the spatial information associated to regressors. Indeed, while sparse priors enforce that only a small subset of variables is used, the assumption that these regressors overlap across all tasks is overly simplistic given the spatial variability observed in real data. In this paper, we propose a convex regularizer for multi-task regression that encodes a more flexible geometry. Our regularizer is based on unbalanced optimal transport (OT) theory, and can take into account a prior geometric knowledge on the regressor variables, without necessarily requiring overlapping supports. We derive an efficient algorithm based on a regularized formulation of OT, which iterates through applications of Sinkhorn's algorithm along with coordinate descent iterations. The performance of our model is demonstrated on regular grids with both synthetic and real datasets as well as complex triangulated geometries of the cortex with an application in neuroimaging.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

1 Introduction

Several regression problems encountered in the high-dimensional regime involve the prediction of one (or several) values using a very large number of regressors. In many of these problems, these regressors relate to physical locations, describing for instance measurements taken at neighboring locations, or, more generally quantities that are tied by some underlying geometry: In climate science, they may correspond to physical measurements (surface temperature, wind velocity) at different locations across the ocean [Chatterjee et al., 2012]; In genomics, they map to positions on the genome [Laurent et al., 2009]; In functional brain imaging, they represent 3D locations in the brain, and a single regression task can correspond to estimating a quantity for a given patient [Owen et al., 2009].

These challenging high-dimensional learning problems have been tackled in recent years using a combination of two approaches: *multitask learning* to increase the sample size and *sparsity*. Indeed, it is not uncommon in these problems to aim at predicting several – not just one – related target variables simultaneously. When considering *multiple regression tasks*, a natural assumption is that prediction functions (and therefore their parameters) for related tasks should share some similarities. This assumption yields the obvious benefit of being able to pool together different datasets to improve the overall estimation of all parameters [Caruana, 1993]. Sparsity has, on the other hand, been a crucial ingredient to help tackle regression problems found for instance in biology or medicine in the "small n large p" regime, where the number of observations n is dominated by the dimension p ($n \ll p$). For such problems, sparsity-promoting regularizations have lead to important successes, both in practice and theory [Tibshirani, 1996, Bickel et al., 2009, Bach et al., 2011], under the collective name of Lasso-type models.

Challenging problems involving regressors tied by some spatial regularity as those mentioned earlier benefit a lot from the combination of both tools. Indeed, when multiple related regression models in the $p \gg n$ regime need to be estimated, a natural assumption is

to consider that each vector of regression coefficients is sparse, and that a common set of active features is shared across all tasks. This intuition has led to several seminal proposals of Lasso-type models, called multi-task Lasso (MTL) or multi-task feature learning (MTFL) [Argyriou et al., 2007, Obozinski and Taskar, 2006]. Both approaches are based on convex ℓ_1/ℓ_2 group-Lasso norms that promote block sparse solutions.

An issue alluded to by Negahban and Wainwright [2008] is that perfect overlap between all tasks can be a too extreme assumption. To understand how to go beyond this binary idea that active coefficients are the same or not, one can notice that in the context of features mapping to physical locations, employing an ℓ_1/ℓ_q norm means assuming that *exactly* the same locations in the physical space, brain or genome are active for each experiment or patient. This is clearly not realistic in several problems [Gramfort et al., 2015].

Our contribution. Our work aims to relax the assumption of perfect overlap across tasks. To do so, we propose to handle non-overlapping supports in standard multi-task models using an *optimal transport metric* between the parameters of our regression models. Optimal transport (OT) has recently gained considerable popularity in signal processing and machine learning problems. This recent outburst of OT applications can be explained by three factors: the inherent ability of OT theory to compute a meaningful distance between probability measures with non-overlapping supports, faster algorithms to compute that metric using entropic regularization [Cuturi, 2013], and their elegant extension to handle non-normalized measures [Chizat et al., 2017] at no additional computational cost. Our convex formulation exploits these strengths and applies them to a more general setting in which we consider (signed) vectors. In practice, our regularized problem is optimized using alternating updates, namely fast proximal coordinate descent and Sinkhorn's algorithm. Sinkhorn iterations are matrix-matrix products which can be sped up on parallel platforms such as GPUs. Our experiments on both synthetic and real data show that our OT model outperforms the state of the art by leveraging the geometrical properties of the regressors.

Related work. To extend ℓ_1/ℓ_q models and relax full overlap assumption, Jalali et al. [2010] proposed to split the regression coefficients into two parts, one common to all tasks and one that is task specific, and to penalize these two parts differently. An ℓ_1 norm is applied on the task-specific part, and an ℓ_1/ℓ_q norm is used on the common part. An alternative proposed by Lozano and Swirszcz [2012] is the *multi-level Lasso* (MLL), which considers instead a product decomposition, with ℓ_1 penalties on both composite variables. Both provide empirical evidence displaying improved performance over block-norm methods. However, experiments show a degraded performance as the overlap between the supports of relevant regressors shrinks. A different approach is proposed by Hernandez-Lobato et al. [2015] where they consider a sparse multi-task regression with outlier tasks and outlier features (non-overlapping features). They introduce a Bayesian model built on a prior distribution with a set of binary latent variables for each feature and each task. Han and Zhang [2015] propose to learn a tree structure on the features, with inner nodes defined as spatially pooled features. The main advantage of this approach is that no assumptions are made on how tasks are related (as in the work of Jawanpuria and Jagarlapudi [2012]). However, the inner nodes will be selected if the supports across tasks do not overlap, resulting in spatially smeared coefficients. Learning how tasks are related adaptively is a potential extension of our work.

This paper is organized as follows. Section 2 introduces our main contribution, the multi-task Wasserstein (MTW) model. We present in Section 3 a computationally efficient optimization strategy to tackle the MTW inference problem. Section 4 demonstrates with multiple experiments the practical benefits of our model compared to Lasso-type models.

Notation. We denote by $\mathbb{1}_p$ the vector of ones in $\mathbb{R}^p$. Given an integer $d \in \mathbb{N}$, $[\![d]\!]$ stands for $\{1, \dots, d\}$. The set of vectors in $\mathbb{R}^p$ with non-negative (resp. positive) entries is denoted by $\mathbb{R}^p_+$ (resp. $\mathbb{R}^p_{++}$). On matrices, log, exp and the division operator are applied element-wise. We use $\odot$ for the element-wise multiplication between matrices or vectors. If X is a matrix, $X_{i\cdot}$ denotes its i^{th} row and $X_{\cdot j}$ its j^{th} column. We define the Kullback-Leibler (KL) divergence between two positive vectors by $\mathrm{KL}(x, y) = \langle x, \log(x/y)\rangle + \langle y - x, \mathbb{1}_p\rangle$ with the continuous extensions $0\log(0/0) = 0$ and $0\log(0) = 0$. We also use the convention that for $x \neq 0$, $\mathrm{KL}(x|0) = +\infty$. The entropy of $x \in \mathbb{R}^n$ is defined as $E(x) = -\langle x, \log(x) - \mathbb{1}_p\rangle$. Finally, for any vector $u \in \mathbb{R}^p$, the support of u is $\mathcal{S}_u = \{i \in [\![p]\!], u_i \neq 0\}$.

2 Multi-task Wasserstein model

Multi-task regression. Consider T datasets of labeled vectors $(X^t, Y^t) \in \mathbb{R}^{n_t \times p} \times \mathbb{R}^{n_t}$, where n_t is the sample size of each set, and p is the dimension of the common space in which all observations lie. Our aim is to estimate, in a high-dimensional regime $n_t \ll p$, T linear regression models:

$$Y^t = X^t\theta^t + \epsilon^t, \quad t \in [\![T]\!] \ ,$$

where $\theta^1, \dots, \theta^T \in \mathbb{R}^p$ are regression coefficients to be estimated from the samples X^t with associated

labels Y^t, and $\epsilon^1, \dots, \epsilon^T \in \mathbb{R}^n$ are additive noise terms assumed to be i.i.d centered Gaussian variables with the same variance $\sigma^2 I_n$. For simplicity, we will assume from now on that $n_t = n$.

Multi-task consensus through Geometric Variance. The idea behind multi-task learning is to estimate $\theta^1, \dots, \theta^T$ jointly, using a regularization term J that promotes some form of similarity between them. All multi-task regression models can then be written:

$$\min_{\theta^1,\dots,\theta^T} \frac{1}{2n} \sum_{t=1}^{T} \|X^t\theta^t - Y^t\|_2^2 + J(\theta^1, \dots, \theta^T) \ . \quad (1)$$

We propose to employ a regularizer that promotes not only sparse solutions, but also some form of "geometric" consensus across all $\theta^1, \dots, \theta^T$ through the use of an arbitrary discrepancy function $\Delta : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, writing $J(\theta^1, \dots, \theta^T) \stackrel{\text{def}}{=} \min_{\bar{\theta} \in \mathbb{R}^p} H(\theta^1, \dots, \theta^T; \bar{\theta})$, where for regularization parameters $\mu \geq 0$ and $\lambda > 0$,

$$H(\theta^1, \dots, \theta^T; \bar{\theta}) \stackrel{\text{def}}{=} \frac{\mu}{T} \overbrace{\sum_{t=1}^{T} \Delta(\theta^t, \bar{\theta})}^{\text{geometric variance}} + \frac{\lambda}{T} \overbrace{\sum_{t=1}^{T} \|\theta^t\|_1}^{\text{sparsity}}, \quad (2)$$

We call the first quantity a geometric variance because it boils down to the usual variance when Δ is the squared Euclidean distance. Indeed, the minimization of $\bar{\theta}$ in J would return the mean of all θ^t, and the first sum in H would then be the variance of these vectors.

An OT Discrepancy for Vectors in $\mathbb{R}^p$. To quantity the geometric variance, we propose to use a new generalized OT metric, that can leverage the fundamental ability of Wasserstein distances to provide a meaningful meta-distance between vectors when a metric on the bins of these vectors is known. However, since OT metrics are defined for positive and normalized vectors, using them in our setting requires some adaptation. Similarly to [Profeta and Sturm, 2018, Mainini, 2012], we propose to split each vector in its positive and negative parts. More formally we write $(x_+, x_-) \in \mathbb{R}_+^p$ such that $x = x_+ - x_-$ by setting $x_+ = \max(x, 0)$ applied elementwise. Next, denoting W the unbalanced Wasserstein distance introduced by Chizat et al. [2017] and described in detail in the next paragraph, we consider in the rest of this work for two arbitrary vectors $x, y \in \mathbb{R}^p$:

$$\Delta(x, y) \stackrel{\text{def}}{=} W(x_+, y_+) + W(x_-, y_-) \ . \quad (3)$$

When $\mu = 0$, (2) boils down to the penalty of T independent Lasso models, one for each task. When the θ^t are fixed, the minimization w.r.t. $\bar{\theta}_+$ (resp. $\bar{\theta}_-$) consists in estimating the barycenter of the θ_+^t (resp. θ_-^t) according to the metric W. When $\lambda = 0$, one forces all the coefficients to be closer according to W.

Unbalanced Wasserstein distance W. The reason why optimal transport distances fit our framework is that they can leverage knowledge on the geometry of regressors, in situations such as those presented in the introduction. In OT, that knowledge is known as a *ground metric*. When working in $\mathbb{R}^p$, this ground metric can be seen as a substitution cost matrix between all p regressors, and is given as a matrix $M \in \mathbb{R}_+^{p \times p}$ of pairwise distances between bins. Following the historical analogy of mass displacement cost, M_{ij} represents the cost to move one unit of mass from location i to location j. In the current context, M may come from the knowledge that features map to certain spatial positions. For instance, in applications where features correspond to positions $(x_1, \dots, x_p)$ in a Euclidean space, the standard cost matrix is given by $M_{ij} = \|x_i - x_j\|_2^2$.

As proposed in [Frogner et al., 2015, Chizat et al., 2017], an optimal transport cost between two nonnegative vectors θ_1 and θ_2 in $\mathbb{R}_+^p$ can be defined by seeking a transport plan $P \in \mathbb{R}_+^{p \times p}$ that: (i) achieves low transport cost $\langle P, M \rangle$; (ii) has marginals $P\mathbb{1}$ (resp. $P^\top \mathbb{1}$) that are as close as possible to θ_1 (resp. θ_2) in KL sense and (iii) has high entropy. These three requirements are reflected in the definition:

$$W(\theta_1, \theta_2) \stackrel{\text{def}}{=} \min_{P \in \mathbb{R}_+^{p \times p}} G(P, \theta_1, \theta_2) \ , \quad (4)$$

where

$$G(P, \theta_1, \theta_2) = \overbrace{\langle P, M \rangle - \varepsilon E(P)}^{\text{transport - entropy}} + \overbrace{\gamma \mathrm{KL}(P\mathbb{1}|\theta_1) + \gamma \mathrm{KL}(P^\top \mathbb{1}|\theta_2)}^{\text{marginal constraints}} \ , \quad (5)$$

and $\varepsilon, \gamma > 0$ are parameters providing a tradeoff between these different objectives.

Large values of $\gamma > 0$ tend to strongly penalize unbalanced transports, and as a result penalize discrepancies between the marginals of P and θ_1, θ_2. The entropy regularization, first introduced by Cuturi [2013], makes the problem strictly convex and computationally faster to solve. A crucial feature of this definition is that the resolution of (4) does not require computing nor storing in memory any optimal plan $P^\star$. Instead, one can study its Fenchel-Rockafellar dual problem given by:

$$W(\theta_1, \theta_2) = \max_{\substack{u,v \\ \in \mathbb{R}_+^p}} \Big[-\varepsilon \langle u \otimes v - 1, K \rangle - \gamma \langle u^{-\frac{\varepsilon}{\gamma}} - 1, \theta_1 \rangle - \gamma \langle v^{-\frac{\varepsilon}{\gamma}} - 1, \theta_2 \rangle \Big], \quad (6)$$

Performing alternating gradient ascent on (6) amounts to computing matrix scalings of a generalized Sinkhorn algorithm (see Section 3).

Well-posedness. We show in this paragraph that a minimizer of (1) exists. To do so, we must prove that the objective function is continuous and coercive.

Lemma 1. *For any* $\theta_1, \theta_2 \in \mathbb{R}_+^p$

$$W(\theta_1, \mathbf{0}) = W(\mathbf{0}, \theta_2) = W(\mathbf{0}, \mathbf{0}) = 0$$

PROOF. We show that $W(\mathbf{0}, \theta_2) = 0$. The result follows directly from the definition of the KL divergence. Let $P \in \mathbb{R}_+^{p \times p}$. We have $\mathrm{KL}(P\mathbb{1}, \mathbf{0}) = 0$ if $P = \mathbf{0}$ and $+\infty$ otherwise. Thus, the minimizer of $G(P, \mathbf{0}, \theta_2)$ is $P^\star = \mathbf{0}$ and we have $W(\mathbf{0}, \theta_2) = G(\mathbf{0}, \theta_1, \mathbf{0}) = 0$. The same reasoning applies to prove $W(\theta_1, \mathbf{0}) = W(\mathbf{0}, \mathbf{0}) = 0$. ■

Proposition 1. The extension (3) preserves the continuity of H at 0.

PROOF. Since H is separable across the (θ^t), we only need to prove that for $\theta, \bar{\theta} \in \mathbb{R}_+^p$ we have $\lim_{(\theta,\bar{\theta})\downarrow 0} W(\theta, \bar{\theta}) = \lim_{\bar{\theta}\downarrow 0} W(0, \bar{\theta}) = \lim_{\theta \downarrow 0} W(\theta, 0) = W(0, 0) = 0$. Let $i, j \in [\![p]\!]$. Suppose $\theta_i, \bar{\theta}_j \neq 0, 0$. G is smooth, convex and coercive w.r.t to P. The first order optimality condition reads:

$$M + \varepsilon \log(P_{ij}) + \gamma \log\left((P\mathbb{1})_i (P^\top \mathbb{1})_j\right) = \gamma \log(\theta_i^t \bar{\theta}_j)$$

$$\Leftrightarrow \exp(M)(P_{ij})^\varepsilon \left((P\mathbb{1})_i (P^\top \mathbb{1})_j\right)^\gamma = (\theta_i^t \bar{\theta}_j)^\gamma$$

When $(\theta, \bar{\theta}) \to (\mathbf{0}, \mathbf{0})$, $(P_{ij})^\varepsilon \left((P\mathbb{1})_i (P^\top \mathbb{1})_j\right)^\gamma \to 0$ and since P is non-negative we have $\forall i, j, P_{ij} \to 0$, *i.e* $P \to \mathbf{0}$. The continuity of G with respect to P leads to $\lim_{(\theta,\bar{\theta})\downarrow 0} W(\theta, \bar{\theta}) = 0$. Lemma 1 guarantees $\lim_{\bar{\theta}\downarrow 0} W(0, \bar{\theta}) = \lim_{\theta \downarrow 0} W(\theta, 0) = W(0, 0) = 0$. ■

Proposition 1 shows that our extension still guarantees that H is continuous at 0. Now we show that the loss function in (1) is coercive.

Proposition 2. The loss function in (1) is coercive.

PROOF. Let's prove that W is bounded from below. Since KL is non-negative, and $\langle P, M \rangle \geq 0$, we have $W(\theta_1, \theta_2) \geq \min_{P \in \mathbb{R}^{p \times p}} -\varepsilon E(P)$ which is minimized at $P^\star_{ij} = 1 \, \forall i, j$. Thus $W(\theta_1, \theta_2) \geq -\varepsilon p^2$. Thus, given that the ℓ_1 norm is non-negative, H is also bounded from below. The coercivity of the loss function follows from the coercivity of the quadratic loss. ■

3 Efficient Optimization of MTW

Loss function. We solve MTW by alternating minimization on the positive and negative parts of the regression coefficients $\boldsymbol{\theta} \stackrel{\text{def}}{=} (\theta^1, \dots, \theta^T)$ and those of $\bar{\theta}$. We will use in what follows bold symbols for sequences of the form $\boldsymbol{z} = (z^1, \dots, z^T)$. Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ denote respectively the optimal transport plans linking $\boldsymbol{\theta}_+$ with $\bar{\theta}_+$ and $\boldsymbol{\theta}_-$ with $\bar{\theta}_-$, and $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ their respective left marginals. Combining (1), (2) and (4), the cost function to minimize is given by:

$$L(\boldsymbol{\theta}; \boldsymbol{P_1}; \boldsymbol{P_2}; \bar{\theta}) = \sum_{t=1}^T \Big[\frac{1}{2n} \|X^t \theta^t - Y^t\|^2 + \frac{\lambda}{T} \|\theta^t\|_1 + \frac{\mu}{T} \left[G(P_1^t, \theta_+^t, \bar{\theta}_+) + G(P_2^t, \theta_-^t, \bar{\theta}_-) \right] \Big] \quad . \quad (7)$$

L is jointly convex in all its variables (since the Kullback-Leibler is jointly convex, proof in Supplementary materials) and the remaining terms are convex and not coupled. The straightforward solution is to minimize L by block coordinate descent. Since the minimization with respect to the variables $(P_1^t, \theta_+^t, \bar{\theta}_+)_t$ and $(P_2^t, \theta_-^t, \bar{\theta}_-)_t$ is similar, we only detail hereafter the minimization with respect to $(P_1^t, \theta_+^t, \bar{\theta}_+)$. The full optimization strategy is provided in Algorithm 1. We alternate with respect to $(\boldsymbol{P_1}, \bar{\theta}_+)$ and each θ_+^t, which can be updated independently and therefore in parallel. We now detail the two steps of the procedure.

Barycenter update. For fixed $\boldsymbol{\theta}_+$, minimizing with respect to $(\boldsymbol{P}_1, \bar{\theta}_+)$ boils down to the unbalanced Wasserstein barycenter computation of [Chizat et al., 2017] which generalizes previous work by Agueh and Carlier [2011] to compute the minimizer of $\min_{\bar{\theta}_+ \in \mathbb{R}_+^p} \frac{1}{T} \sum_{t=1}^T W(\theta_+^t, \bar{\theta}_+)$. This is equivalent to minimizing simultaneously in $P_1^1, \dots, P_1^t \in \mathbb{R}_+^{p \times p}$ and $\bar{\theta}_+ \in \mathbb{R}_+^p$ the objective:

$$\varepsilon \sum_{t=1}^T \mathrm{KL}(P_1^t, K) + \gamma \sum_{t=1}^T \mathrm{KL}(P_1^t \mathbb{1} | \theta_+^t) + \gamma \sum_{t=1}^T \mathrm{KL}({P_1^t}^\top \mathbb{1} | \bar{\theta}_+) \quad . \quad (8)$$

As pointed out by Chizat et al. [2017] and recalled in (6), Fenchel-Rockafellar duality allows to minimize over dual variables $u^t, v^t \in \mathbb{R}^p$ instead of considering plans $P_1^t \in \mathbb{R}_+^{p \times p}$. P_1^t can be recovered as $(u_i^t K_{ij} v_j^t)_{ij}$ and its left marginal, needed for the coefficient update, is given by $m_1^t \stackrel{\text{def}}{=} P_1^t \mathbb{1} = u^t \odot K v^t$. These steps are summarized in Alg. 4. We monitor the largest relative change of barycenter $\bar{\theta}_+$ to stop our loop.

Coefficients update. Minimizing with respect to one θ_+^t while keeping all other variables fixed to their current estimate yields problem (9), where the ℓ_1 penalty becomes linear due to the positivity constraint. Given the left marginal m_1, the problem reads for all θ_+^t (omitting index t):

$$\min_{\theta_+ \in \mathbb{R}_{++}^p} \Big[\frac{1}{2n} \|X\theta_+ - X\theta_- - Y\|^2 + \sum_{i=1}^p \frac{\mu\gamma}{T} (\theta_{+i} - m_i \log(\theta_{+i})) + \lambda \theta_{+i} \Big] \quad . \quad (9)$$

Algorithm 1 Alternating optimization

Input: θ^0, hyperparameters: $\mu, \epsilon, \gamma, \lambda$ and M.
Output: $\boldsymbol{\theta}$, the minimizer of (1).
repeat
  for $t = 1$ **to** T **do**
    Update θ^t_+ with proximal coordinate descent.
    Update θ^t_- with proximal coordinate descent.
  end for
  Update the left (resp. right) marginals $m^1_+, \dots, m^t_+$ and $\bar{\theta}_+$ resp($m^1_-, \dots, m^t_-$ and $\bar{\theta}_-$) with generalized Sinkhorn.
until convergence

Algorithm 2 Generalized Sinkhorn[Chizat et al., 2017]

Input: $\theta^1, \dots, \theta^T$
Output: Wasserstein barycenter of $\theta^1, \dots, \theta^T$ and marginals $m^1, \dots, m^T$.
Initialize for $(t = 1, \dots, T)$ $(u^t, v^t) = (\mathbb{1}, \mathbb{1})$,
repeat
  for $t = 1$ **to** T **do**
    $u^t \leftarrow (\theta^t / Kv^t)^{\frac{\gamma}{\gamma+\varepsilon}}$
  end for
  $\bar{\theta} \leftarrow \left(\frac{1}{T}\sum_{t=1}^T (v^t \odot K^\top u^t)^{\frac{\varepsilon}{\varepsilon+\gamma}}\right)^{\frac{\varepsilon+\gamma}{\varepsilon}}$
  for $t = 1$ **to** T **do**
    $v^t \leftarrow \left(\bar{\theta} / K^\top u^t\right)^{\frac{\gamma}{\gamma+\varepsilon}}$
  end for
until convergence
for $t = 1$ **to** T **do**
  $m^t = u^t \odot Kv^t$
end for

The penalty is a separable sum of convex functions with tractable proximal operators, and therefore (9) can be solved by proximal coordinate descent [Tseng, 2001, Fercoq and Richtárik, 2015]. The following proposition, proved in the appendix, gives a closed-from solution for that proximal operator.

Proposition 3. Let $a, b, \alpha \in \mathbb{R}_{++}$. Function $g : x \mapsto (x - a\log(x)) + bx$ is convex on $\mathbb{R}_{++}$, and one has:

$$\text{prox}_{\alpha g}(y) = \frac{1}{2}\left[-\alpha(b+1) + y + \sqrt{(\alpha(b+1) - y)^2 + 4\alpha a}\right]$$

Entropy regularization. While large values of ε (strong entropy regularization) induce undesired blurring, low values tend to cause a well-documented numerical instability [Chizat et al., 2017, Schmitzer, 2016], which can be avoided by moving to the log-domain [Schmitzer, 2016]. Also for experiments performed on regular grids such as images, one should leverage the separability of the kernel K as proposed in [Solomon et al., 2015] to recover far more efficient implementations. This also applies to log-domain computations [Schmitz et al., 2017]. We use in this work these crucial improvements over naive implementations of Sinkhorn algorithms.

Accelerating convergence with warm-start. To speed up convergence, we initialize the Sinkhorn scaling vectors to their previous values, kept in memory between two barycenter computations. This does not affect convergence because of the convexity of the objective function. Note that transport plans $P^1, \dots, P^T$ are never instantiated, as this would be too costly. We only compute their left marginals $m^1, \dots, m^T$, which are involved in the coefficients update. We track both the relative evolution of the objective function and that of the norm of the coefficients to terminate the algorithm. Performing less Sinkhorn iterations per barycenter update yields in practice faster convergence, while reaching the same final tolerance threshold. See supplementary materials for an illustration of this tradeoff.

Hyperparameter tuning. The MTW model has four hyperparameters: $\epsilon, \gamma, \mu, \lambda$. We provide in this section practical guidelines to set parameters ε and γ within the unbalanced Wasserstein distance.

Setting ε. As mentioned above, entropy regularization speeds up computations but induces blurring. In our experiments we observe that a value of $1/sp$, where s is the median of the ground metric M, provides an excellent tradeoff between speed and performance.

Setting γ. In the barycenter definition (8), γ controls the influence of the marginals: as γ goes to 0, P tends to K since we can ignore marginal constraints. This transport plan, however, only leads to a local blur with no transport, so that the mass of the barycenter $\bar{\theta}\mathbb{1} \to 0$. To avoid this degenerate behavior, consider the case where $\gamma \gg \varepsilon$ so entropy regularization can be neglected in (8). The corresponding approximate objective function is given by:

$$\sum_{t=1}^T \left[\langle P^t, M\rangle + \gamma \text{KL}(P^t\mathbb{1}|a^t) + \gamma \text{KL}({P^t}^\top\mathbb{1}|a)\right]. \quad (10)$$

Deriving the first order conditions, for any $t \in [\![T]\!]$:

$$M_{ij} + \gamma \log\left(\frac{P^t_{i.}P^t_{.j}}{a^t_i \bar{a}_j}\right) = 0 \quad \text{and} \quad \bar{a} = \frac{1}{T}\sum_{t=1}^T {P^t}^\top\mathbb{1},$$

By combining the two, we get for any $\tau \in [0, 1]$:

$$\gamma \geq -\frac{\max M}{\log \tau} \Rightarrow \bar{\psi} \geq \tau\left(\frac{1}{T}\sum_{t=1}^T \sqrt{\psi_t}\right)^2, \quad (11)$$

where $\bar{\psi}, \psi_1, \dots, \psi_t$ denote the respective masses of

$\bar{a}, a_1, \ldots, a^T$ i.e $\psi^t = {a^t}^\top \mathbb{1}$. Therefore, (11) provides an adaptive parametrization of γ that guarantees a lower bound on the mass of $\bar{\theta}$ as a fraction of the $\ell_{0.5}$ pseudo-norm of those of the inputs. In practice, in all experiments we use $\tau = 0.5$ and set $\gamma = \tau \left(\frac{1}{T} \sum_{t=1}^{T} \sqrt{\psi_t} \right)^2$]. Code in Python can be found in https://github.com/hichamjanati/mtw.

With ε and γ fixed, only two hyperparameters (μ, λ) remain. These control respectively the similarity between tasks and sparsity. Setting two parameters is not more than what is required by Dirty models [Jalali et al., 2010] or an Elastic-Net.

4 Experiments

Benchmarks. To quantify the benefit of multi-task inference, we use a Lasso estimator independently run on each task as a standard baseline. We compare the performance of our algorithm against *Dirty models* [Jalali et al., 2010] and *Multi-level Lasso* [Lozano and Swirszcz, 2012]. When the ground truth is available, we evaluate support identification using the area under the curve (AUC) of precision-recall.

The Group Lasso learning model [Argyriou et al., 2007, Obozinski and Taskar, 2006] (a.k.a MTFL) can be expressed by setting the penalty J to be an ℓ_1/ℓ_2 mixed norm: $\|\boldsymbol{\theta}\|_{21} = \sum_{j=1}^{p} \sqrt{\sum_{t=1}^{T} (\theta_j^t)^2}$. Such a regularization forces all the θ^t to have the exact same support, $\mathcal{S}_{\theta^t} = \mathcal{S}_{\theta^{t'}}$ for all t, t'. To nuance this very strong assumption, Dirty models [Jalali et al., 2010] propose to decompose $\theta^t = \theta_c^t + \theta_s^t$, where θ_c^t is common between all tasks (i.e $\mathcal{S}_{\theta_c^t} = \mathcal{S}_{\theta_c^{t'}} \forall t, t'$) and θ_s^t is specific to each one. The regularization then writes:

$$J_{\text{Dirty}}(\boldsymbol{\theta}) = \mu \|(\theta_c^1, \ldots, \theta_c^T)\|_{21} + \lambda \sum_{t=1}^{T} \|\theta_s^t\|_1 \ . \quad (12)$$

When $\theta_s = 0$ (resp. $\theta_c = 0$) one falls back to a Group Lasso (resp. independent Lasso) estimator [Argyriou et al., 2007, Obozinski and Taskar, 2006].

Multi-level Lasso (MLL) applies instead the ℓ_1 penalty on two levels of a product decomposition $\theta_j^t = C_j S_j^t$ where $C \in \mathbb{R}^p$ is common across tasks and $S^t \in \mathbb{R}^p$ is task specific. For the model to be identifiable, C is constrained to be non-negative. The (MLL) penalty:

$$J_{\text{MLL}}(S^1, \ldots, S^T; C) = \mu \|C\|_1 + \frac{\lambda}{T} \sum_{t=1}^{T} \|S^t\|_1 \ . \quad (13)$$

As shown by Lozano and Swirszcz [2012], (13) is equivalent to a standard multi-task regression problem with the non-convex regularization:

$$J(\boldsymbol{\theta}) = \frac{1}{T} \sum_{j=1}^{p} \sqrt{\sum_{t=1}^{T} |\theta_j^t|} \ . \quad (14)$$

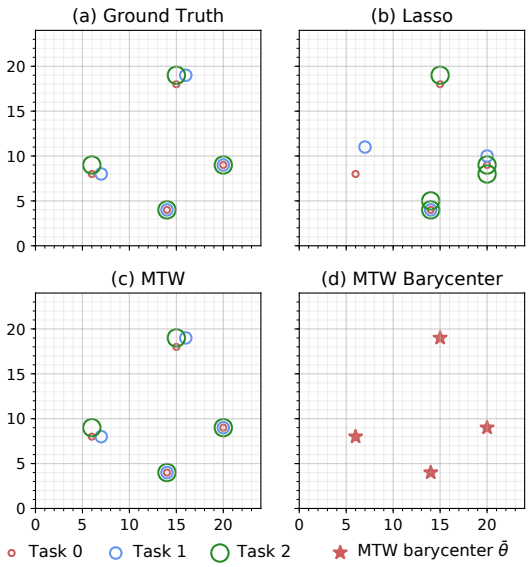


Figure 1: 3 sets of color-labeled regression coefficients on a 2D grid. Each circle represents a non-zero coefficient. Different radii are used for a better distinction of overlapping features. **(a)** Inputs. Joint estimation of 3 ill-posed regression tasks using: **(b)** Lasso **(c)** MTW, latent Wasserstein barycenter shown in **(d)**.

4.1 Synthetic data

We simulate 3 coefficients $(\theta^t)_{t=1\ldots3}$ defined on a 2D grid of shape (24×24), and that each vector of coefficients is 4-sparse: each has only 4 non-zero values (see Figure 1). Each coefficient can be seen as a 24×24 image. We thus have 3 tasks with $p = 576$. The design matrix is obtained by applying a Gaussian filter to the image with standard deviation of 1 pixel, and downsampling the blurred image by taking the mean over (4 $\times$ 4) blocks. This leads to $n = 36$ samples. We set the Gaussian noise variance σ^2 so that the signal-noise-ratio (SNR) is equal to 3, with $\text{SNR}^2 \stackrel{\text{def}}{=} \sum_t \|X^t \theta^t\|_2^2 / (T\sigma^2)$.

To control the overlap ratio between the supports and guarantee their proximity, we first start by selecting two random pixels and randomly translating the non-overlapping features by a one or two pixels for the corresponding tasks. The coefficient values are drawn uniformly between 20 and 30.

Here coefficients map to image pixels, so we employed the MTW with a non-negativity constraint ($\theta^t \in \mathbb{R}_+$). Figure 2 shows the distribution of the best AUC scores of 100 experiments (different coefficients and noise). As expected, independent Lasso estimators do not benefit

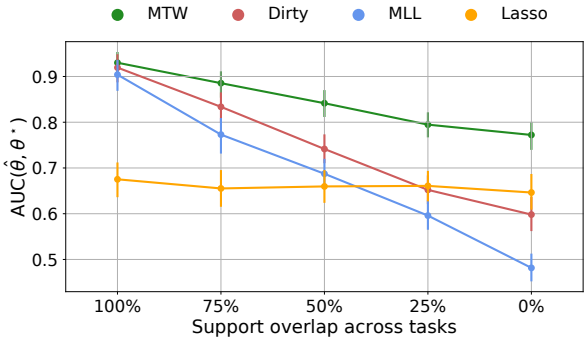


Figure 2: Mean AUC score (100 runs) cf the estimated coefficients versus ground truth with Dirty models, the Multi-level Lasso (MLL), independent Lasso estimators, and Multi-task Wasserstein (MTW).

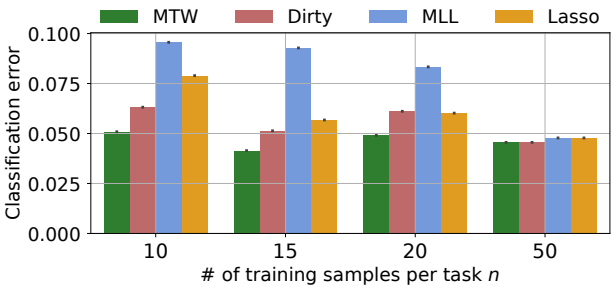


Figure 3: Classification error on a left-out validation set. MTW: Multi-task Wasserstein. MLL: Multi-level Lasso. MTW outperforms all methods as n decreases. Black bars show 95% confidence intervals over 50 different random splits of the data.

from task relatedness. Yet, they perform better than Dirty and MLL when supports poorly overlap, which confirms the results of Negahban and Wainwright [2008]. MTW however clearly wins in all scenarios.

4.2 Handwritten digits recognition.

We use the dataset of van Breukelen et al. [1998] consisting of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. We select 6 tasks corresponding to the digits ('0'–'5') and the features corresponding to the pixel averages of (2×3) windows of the original (unprovided) (30×48) handwritten digit images, thus $p = 240$. We set $n = 10$; 15; 20 or 50 training samples per task. Model selection is carried out using a 5-folds cross-validation. We report in figure 3 the mean misclassification rate on the left-out validation set containing $n_v = 200 - n$ samples per task for 50 different random splits of the training / validation data. MTW is particularly efficient in the small n regime with a significant 95% confidence interval. The regression coefficients obtained by each model are displayed in the appendix.

4.3 MEG source localization.

We use the publicly available dataset DS117 of Wakeman and Henson [2015]. DS117 contains MEG and EEG recordings of 16 subjects who underwent the same cognitive visual stimulus consisting in pictures of: famous people; scrambled faces; unfamiliar faces. Using the provided MRI scans, we compute the design matrices X^t i.e the forward operators of the magnetic field generated by a cortical triangulation of $p = 2101$ locations using the MNE software [Gramfort et al., 2013]. The regression outputs Y^t correspond to measurements of the magnetic field on the surface of the scalp recorded by $n = 204$ sensors (we keep only MEG gradiometers), as for example used in [Owen et al., 2009]. Since the true brain activations $\theta^\star$ are unknown, we quantify the performance of our model using the real $(X^t)_{t=1,\dots,T}$ and simulated $(Y^t)_{t=1,\dots,T}$. Note that the assumption of partial overlap is particularly adapted to this application. Indeed, while functional organization of the brain is comparable between subjects at a certain scale, one cannot assume that the activation foci are perfectly overlapping between individuals. In other words, active brain regions tend to be close in the population but not identical [Thirion et al., 2007, Xu et al., 2009].

Simulated activations. The regression coefficients (sources) are k-sparse ($k \in [\![11]\!]$), *i.e* all zero except for k random locations chosen respectively in one of 11 distinct brain regions (displayed in supplementary material). Their amplitudes are taken uniformly within $20 - 30\,\text{nAm}$. Their sign is then decided by a coin toss (Bernoulli with 0.5 parameter). We generate in this manner a set of different regression coefficients for the number of tasks desired. We construct the outcome Y^t with a SNR equal to 4. For MTW, the ground metric M is the distance on the cortical mesh of p vertices. It corresponds to the geodesic distance on the complex topology of the cortex.

Illustrative example. MTW is expected to be most valuable for non-overlapping supports. To visually illustrate the benefits of our model, we randomly select 2 subjects and simulate regression coefficients with 3 sources per task with only one common feature. Figure 4 shows MTW at its best: MTW leverages the geometrical proximity of the sources and thereby perfectly recovers the true supports. The independent Lasso estimator however reaches a poor AUC of 0.54. It selects features very far from the true brain regions which can lead to erroneous conclusions. Moreover, the

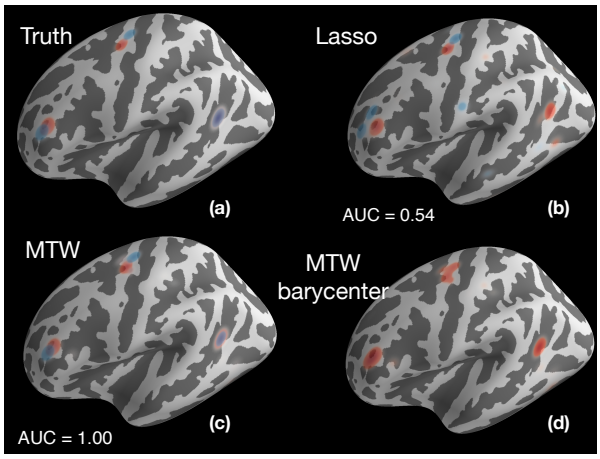


Figure 4: Each color corresponds to one of the two subjects (except for (d)). **(a)**: True sources: one common feature (right side of the displayed hemisphere) and two non-overlapping sources. **(b, c)**: Sources estimated by (b) Lasso and (c) MTW with the highest AUC score. (d) Shows the barycenter $\bar{\theta}$ associated with MTW model. In this figure, the displayed activations were smoothed for the sake of visibility.

latent barycenter $\bar{\theta}$ highlights the most representative sources of the cohort of subjects studied (Fig. 4 (d)).

Effect of degree of overlap. Using 3 subjects, we perform 30 trials with different noise and coefficients locations and values (Figure 5). We make the localization even harder by selecting 5 sources, *i.e.* 5 non-zero features per task. We select the best performance of all models in terms of AUC score. MTW outperforms all benchmarks in recovering the true supports in all scenarios. Unlike Dirty models, MLL fails to recover perfectly overlapping supports and has a large variance. This behavior may be due to the non-convexity of the penalty in (14) and potentially bad local minima.

Effect of number of tasks. When the number of non-zero features increases, recovering the support is more difficult. Figure 6 shows that MTW handles particularly well that scenario, as tasks increase. We compute the mean AUC score of 20 trials for 2, 4 and 8 tasks, 2 to 6 non-zero coefficients with an overlap of supports set to 50% and a SNR equal to 4. The curves obtained by independent Lasso overlap as it does not benefit from additional tasks. Dirty models handle relatedness through the ℓ_1/ℓ_2 penalty which only improves the estimation of the common features across tasks. This explains why the performance of Dirty models with 4 and 8 tasks is the same. MTW is unique in that it benefits from all 8 tasks.

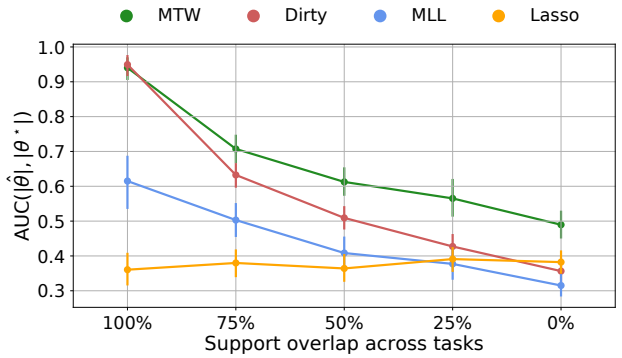


Figure 5: Comparaison of different values of supports overlap using 3 tasks. Mean AUC score obtained on MEG data simulation with a SNR = 4 over 30 different experiments. MTW: Multi-task Wasserstein. MLL: Multi-level Lasso. MTW outperforms other models for all supports overlap fractions.

Figure 6: Mean AUC score for different numbers of tasks and support sizes with an overlap of 50% (20 different runs). MTW benefits more from additional tasks; obtained on MEG data simulation. SNR = 4.

Conclusion

The seminal work of Caruana [1993] has motivated a series of contributions leveraging the presence of multiple and related learning tasks (MTL) to improve statistical performance. Our work is one of them in the context of sparse high dimensional regression tasks where regressors can be associated to a geometric space. Using Optimal Transport to model proximity between coefficients, we proposed a convex formulation of MTL that does not require any overlap between the supports, contrarily to previous literature. We show how our Multi-task Wasserstein (MTW) model can be solved efficiently relying on proximal coordinate descent and Sinkhorn's algorithm. Our experiments on synthetic and real data demonstrate that regardless of overlap, MTW leverages the geometry of the problem to outperform standard multi-task regression models.

Acknowledgments

MC and HJ acknowledge the support of a *chaire d'excellence de l'IDEX Paris Saclay*. AG and HJ were supported by the European Research Council Starting Grant SLAB ERC-YStG-676943.

References

S. Chatterjee, K. Steinhaeuser, A. Banerjee, S. Chatterjee, and A. Ganguly. Sparse group lasso: Consistency and climate applications. In *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, pages 47–58, 12 2012. ISBN 9781611972320.

J. Laurent, O. Guillaume, and J-P Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553431.

J. Owen, H. T. Attias, K. Sekihara, S. S. Nagarajan, and D. P. Wipf. Estimating the location and orientation of complex, correlated neural activity using MEG. In *NIPS*, pages 1777–1784. 2009.

R. Caruana. Multitask Learning: A Knowledge-Based Source of Inductive Bias. *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 08 2009.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. 2007.

G. Obozinski and B. Taskar. Multi-task feature selection. In *ICML. Workshop of structural Knowledge Transfer for Machine Learning*, 2006.

S. Negahban and M. J. Wainwright. Joint support recovery under high-dimensional scaling: Benefits and perils of $l1,\infty$ regularization. *NIPS*, 2008.

A. Gramfort, G. Peyré, and M. Cuturi. Fast optimal transport averaging of neuroimaging data. In *IPMI 2015*, July 2015.

M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *NIPS*, 2013.

L. Chizat, G. Peyré, B. Schmitzer, and F-X. Vialard. Scaling Algorithms for Unbalanced Transport Problems. *arXiv:1607.05816 [math.OC]*, 2017.

A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A Dirty Model for Multi-task Learning. *NIPS*, 2010.

A. Lozano and G. Swirszcz. Multi-level Lasso for Sparse Multi-task Regression. *ICML*, 2012.

D. Hernandez-Lobato, J. Miguel Hernandez-Lobato, and Z. Ghahramani. A probabilistic model for dirty multi-task feature selection. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1073–1082. PMLR, 07–09 Jul 2015.

L. Han and Y. Zhang. Learning tree structure in multi-task learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 397–406. ACM, 2015. ISBN 978-1-4503-3664-2. doi: 10.1145/2783258.2783393.

Pratik Jawanpuria and Sakethanath Jagarlapudi. A convex feature learning formulation for latent task structure discovery. 06 2012.

Angelo Profeta and Karl-Theodor Sturm. Heat flow with dirichlet boundary conditions via optimal transport and gluing of metric measure spaces. 2018.

E. Mainini. A description of transport cost for signed measures. *Journal of Mathematical Sciences*, 181(6): 837–855, Mar 2012. ISSN 1573-8795. doi: 10.1007/s10958-012-0718-2.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM*, 43(2):904–924, 2011.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.

O. Fercoq and P.s Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM Journal on Optimization*, 25:1997–2023, 2015.

B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. 2016.

J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4):66:1–66:11, July 2015. ISSN 0730-0301. doi: 10.1145/2766963.

M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngolè, and D. Coeurjolly. Wasserstein Dictionary Learning: Op-

timal Transport-based unsupervised non-linear dictionary learning. Working Papers 2017-84, Center for Research in Economics and Statistics, August 2017.

M. van Breukelen, R.P.W. Duin, D.M.J. Tax, and J.E. den Harto. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

D.G. Wakeman and R.N.A. Henson. A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, 2(150001), 2015. doi: 10.1038/sdata.2015.1.

A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86, 10 2013.

Bertrand Thirion, Alan Tucholka, Merlin Keller, Philippe Pinel, Alexis Roche, Jean-François Mangin, and Jean-Baptiste Poline. High level group analysis of fmri data based on dirichlet process mixture models. In Nico Karssemeijer and Boudewijn Lelieveldt, editors, *Information Processing in Medical Imaging*, pages 482–494, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-73273-0.

Lei Xu, Timothy D. Johnson, Thomas E. Nichols, and Derek E. Nee. Modeling inter-subject variability in fmri activation location: A bayesian hierarchical spatial model. *Biometrics*, 65(4):1041–1051, 2009. doi: 10.1111/j.1541-0420.2008.01190.x.

This appendix is organized as follows. Section A presents details on MTW: convexity, proximal coordinate descent and some background on Sinkhorn's algorithm where we discuss a log-stabilized version Schmitzer [2016] that is used in all our experiments. Section B provides mathematical details on tuning the hyperparameters of Dirty models. Section C provides further details on model selection and experiments. Finally, section D provides the Python code used in our experiments.

A Technical details on MTW

Joint convexiy. Recall the loss function:

$$L(\boldsymbol{\theta}; \boldsymbol{P_1}; \boldsymbol{P_2}; \bar{\theta}) = \sum_{t=1}^{T} \Big[\frac{1}{2n} \|X^t \theta^t - Y^t\|^2 + \frac{\lambda}{T} \|\theta^t\|_1 + \frac{\mu}{T} \left[G(P_1^t, \theta_+^t, \bar{\theta}_+) + G(P_2^t, \theta_-^t, \bar{\theta}_-) \right] \Big] \ .$$

where

$$G(P, \theta_1, \theta_2) = \overbrace{\langle P, M \rangle - \varepsilon E(P)}^{\text{transport - entropy}} + \overbrace{\gamma \mathrm{KL}(P\mathbb{1}|\theta_1) + \gamma \mathrm{KL}(P^\top \mathbb{1}|\theta_2)}^{\text{marginal constraints}} \ ,$$

The quadratic loss function and the ℓ_1 penalty are convex and separable across the $(\theta^t)_t$. The transport and entropy terms in G are convex and separable across the $(P^t)_t$. The only coupled terms involved in L are the marginal constraints in G. To prove joint convexity of L we only need to prove that of KL (since taking out the marginal is a linear operator).

Let $x, y \in \mathbb{R}_+^p$. We defined the Kullback-Leibler function as:

$$\mathrm{KL}(x, y) = \sum_{i=1}^{p} x_i \log(x_i / y_i) + y_i - xi$$

Since KL is an element-wise sum, all we need to show is the joint convexity of $f : (a, b) \mapsto a \log(a/b)$ in $\mathbb{R}_+^2$.

Let $\tau \in [0, 1]$ and $a_1, a_2, b_1, b_2 > 0$. Denote $a_\tau = \tau a_1 + (1 - \tau) a_2$ and $b_\tau = \tau b_1 + (1 - \tau) b_2$. And let $g : x \mapsto x \log(x)$.

g is convex. Using Jensen's inequality:

$$\begin{aligned} f(a_\tau, b_\tau) &= a_\tau \log(a_\tau / b_\tau) \\ &= b_\tau g(a_\tau / b_\tau) \\ &= b_\tau g\left(\frac{\tau b_1}{b_\tau} \frac{\tau a_1}{\tau b_1} + \frac{(1-\tau) b_2}{b_\tau} \frac{(1-\tau) a_2}{(1-\tau) b_2} \right) \\ &\leq b_\tau \left(\frac{\tau b_1}{b_\tau} g\left(\frac{\tau a_1}{\tau b_1} \right) + \frac{(1-\tau) b_2}{b_\tau} g\left(\frac{(1-\tau) a_2}{(1-\tau) b_2} \right) \right) \\ &= \tau b_1 g\left(\frac{a_1}{b_1} \right) + (1-\tau) b_2 g\left(\frac{a_2}{b_2} \right) \\ &= \tau f(a_1, b_1) + (1-\tau) f(a_2, b_2) \end{aligned}$$

Therefore, f is jointly convex. □

Coordinate descent. Recall that the optimization problem solved by our estimator MTW is carried out by alternating between independent coefficients updates and a barycenter computation. First, we give a proof for Proposition 3.1 just recall here:

Proposition 4. Let $a, b \in \mathbb{R}_+$. The function $g : x \mapsto (x - a \log(x)) + bx$ is convex and proximable on $\mathbb{R}_{++}$, moreover its proximal operator is given by:

$$\mathrm{prox}_{\alpha g}(y) = \frac{1}{2} \left[-\alpha(b+1) + y + \sqrt{(\alpha(b+1) - y)^2 + 4\alpha a} \right]$$

Proof. g is clearly convex. Its proximal operator, defined on $\mathbb{R}_{++}$, is given by the minimizer of the problem:

$$\begin{aligned} \mathrm{prox}_{\alpha g}(y) &= \min_x \frac{1}{2}(x - y)^2 + \alpha g(x) \\ &= \min_x \frac{1}{2}(x - y)^2 + -\alpha a \log(x) + \alpha(b+1)x \end{aligned}$$

The objective function above is differentiable, strictly convex and goes to $+\infty$ when $x \to 0^+$ or $x \to +\infty$. Thus, its minimizer is unique and is the solution of the necessary first order optimality condition:

$$\begin{aligned} x - y - \frac{\alpha a}{x} + \alpha b + \alpha &= 0 \\ \Rightarrow x^2 + \alpha(b+1) - yx - \alpha a &= 0 \end{aligned}$$

The positive solution of the quadratic equation above is given by $x = \frac{1}{2}\left[-\alpha(b+1) + y + \sqrt{(\alpha(b+1) - y)^2 + 4\alpha a} \right]$. ∎

Now recall the coefficient update problem:

$$\min_{\theta \in \mathbb{R}_{++}^p} \frac{1}{2n} \|X^t \theta - Y^t\|^2 + \sum_{i=1}^{p} \frac{\mu\gamma}{T} (\theta_i - P_{i\cdot} \mathbb{1} \log(\theta_i)) + \lambda \theta_i \tag{A.1}$$

Algorithm 3 Proximal coordinate descent

Input: X^t, Y^t, α, P, descent steps $\eta_j = \frac{1}{\sum_{i=1}^n {X^t}_{ij}^2}$
Initialize for $\theta = \theta_0$
repeat
  for $j = 1$ **to** p **do**
    $\theta_j = \text{prox}_{\alpha g_j}\left(\theta_j - \eta {X^t_{\cdot j}}^\top (X^t - Y^t)\right)$
  end for
until convergence

Which can be rewritten as:

$$\min_{\theta \in \mathbb{R}^p_{++}} \frac{1}{2n}\|X^t\theta - Y^t\|^2 + \alpha \sum_{i=1}^p g_i(\theta_i) \tag{A.2}$$

Where $g_i : x \mapsto (x - a_i \log(x)) + bx$ with $\alpha = \frac{\mu\gamma}{T}$, $a = P\mathbb{1}$ and $b = \frac{\lambda T}{\gamma\mu}$.

Computing the proximal operator of $G = \sum_i g_i$ boils down to carrying out the proximal operators $\text{prox}_{\alpha g_i}$, element-wise. Therefore, problem (A.2) can be solved using proximal coordinate descent Fercoq and Richtárik [2015] (Algorithm 3).

Sinkhorn's algorithm. The generalized Sinkhorn algorithm used to compute the Unbalanced Wasserstein barycenter may suffer from numerical as instability as the entropy regularization goes to zero i.e when $\epsilon \to 0$. As recalled in Algorithm 4, the barycenter update requires taking the power $\frac{\gamma+\epsilon}{\epsilon}$ of the transport marginals. Typically for the value of $\epsilon = \frac{1}{mp}$ where m is the median value of the cost matrix M, we encounter overflow errors for a certain range of hyperparameters. To allievate this problem, we rely on the log-stabilized version first introduced by Schmitzer [2016]. Consider the change of variables $u' = u' \exp(a), v' = v' \exp(b)$. The idea is to absorbe the large values of the scaling variables in log-domain (i.e a and b) while keeping u' and v' close to 1 as possible. We rely on this trick and allow our model to automatically switch to log-stabilized Sinkhorn when numerical errors are met.

For simulations with synthetic images, we apply the Kernel matrix $\exp(-M/\epsilon)$ using fast convolutions which reduces considerably the complexity of the algorithm Solomon et al. [2015]. Indeed, since our cost matrix M is simply a separable euclidean distance over a square grid, applying the Kernel K to an image is equivalent to computing convolutions its rows and then the columns of the obtained image. Moreover, this kernel separability property still be exploited in log-domain Schmitz et al. [2017].

Alternating optimization. As discussed in section 3, the minimized loss is jointly convex. We observe that in practice, performing a few tens of iterations of Sinkhorn speeds up the convergence. This trade-off is illustrated in Figure A.1 where we show the optimality gap of the loss function w.r.t to different numbers of iterations of Sinkhorn updates. For proximal coordinate descent however, we wait for convergence in each inner loop.

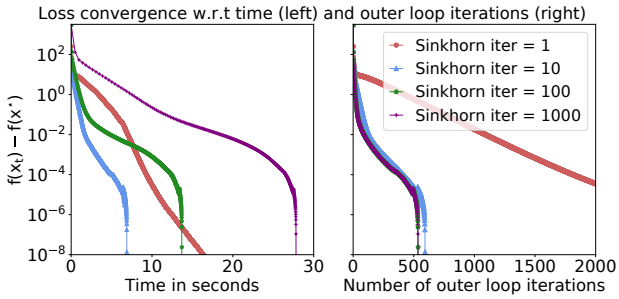


Figure A.1: Illustration of alternating optimization trade-off.

B Dirty models

In this section we show that for Dirty models, hyper-parameters need not to be tuned over a 2D grid but within a surface between the lines with slopes 1 and $\frac{1}{\sqrt{T}}$ where T is the number of tasks. Recall the optimization solved by Multi-task Dirty models with ℓ_1/ℓ_2 norms:

$$\min_{\substack{\theta^1, \theta^2 \\ \in \mathbb{R}^{p \times T}}} \sum_{t=1}^T \frac{1}{2n}\|X^t\theta_c^t + X^t\theta_s^t - Y^t\|^2 + \mu\|\Theta_c\|_{2,1} + \lambda\|\Theta_s\|_1 \ , \tag{B.1}$$

Let's denote the column stacking $\Theta = [\theta^1, \dots, \theta^T]$ and similarly the block diagonal matrix $\boldsymbol{X} = \text{diag}(X^1, \dots, X^T)$ and $\boldsymbol{Y} = \boldsymbol{X}\Theta$.

Algorithm 4 Generalized Sinkhorn Chizat et al. [2017]

Input: $\theta^1, \dots, \theta^T$
Initialize for $(t = 1, \dots, T)$ $(u^t, v^t) = (\mathbb{1}, \mathbb{1})$,
repeat
  for $t = 1$ **to** T **do**
    $u^t \leftarrow \left(\frac{\theta^t}{Kv^t}\right)^{\frac{\gamma}{\gamma+\varepsilon}}$
  end for
  $\bar{\theta} \leftarrow \left(\frac{\sum_{t=1}^T (vK^\top u^t)^{\frac{\varepsilon}{\varepsilon+\gamma}}}{T}\right)^{\frac{\varepsilon+\gamma}{\varepsilon}}$
  for $t = 1$ **to** T **do**
    $v^t \leftarrow \left(\frac{\bar{\theta}}{K^\top u^t}\right)^{\frac{\gamma}{\gamma+\varepsilon}}$
  end for
until convergence

The optimality condition for problem (B.1) reads:

$$0 \in \boldsymbol{X}^\top(\boldsymbol{X}\Theta_c^* + \boldsymbol{X}\Theta_s^* - \boldsymbol{Y}) + \mu\partial_{\ell_{21}}(\Theta_c^*) + \lambda\partial_{\ell_1}(\Theta_s^*)$$

The subdifferential of ℓ_{21} is simply the projection over the unit ball of its dual norm $\ell_{2\infty}$ at $\Theta \neq 0$ and is the set of all elements of that ball otherwise. Thus, for Θ^* equal to 0 we get:

$$\|\boldsymbol{X}^\top\boldsymbol{Y}\|_{2\infty} \leq \mu$$
$$\|\boldsymbol{X}^\top\boldsymbol{Y}\|_{\infty} \leq \lambda$$

The bounds above define a rectangular box over which the gridsearch must be performed. However, we can show that this gridsearch can be reduced to a much smaller triangle.

Suppose $\exists(j,k)$ s.t $\Theta_s^{j,k} \neq 0$. Therefore

$$\exists Z_c \in \mu\partial_{\ell_{21}}(\Theta_c^*) \quad \mu|Z_c^{j,k}| = \lambda$$
$$\Rightarrow \quad \mu \geq \lambda$$

Thus, when $\lambda > \mu$, the model reduces to an independent Lasso estimator.

Now suppose $\exists(j,k)$ s.t. $\Theta_c^{j,k} \neq 0$. Therefore

$$\exists Z_s \in \mu\partial_{\ell_1}(\Theta_s^*) \quad \mu\frac{\Theta_s^{j,k}}{\|\Theta_s^j\|_2} = \lambda Z_s^{j,k}$$
$$\Rightarrow \quad \mu \leq \sqrt{T}\lambda$$

Thus, when $\sqrt{T}\lambda < \mu$, the model reduces to a group-Lasso estimator.

C Simulation details

model selection. For all simulations, we selected the best hyperparameters of each model among a set of hyperparameters set as follows. For Lasso, we set a logarithmic scale of 100 values between $\mu_{\max} = \|\boldsymbol{X}^\top Y\|_\infty$ and $\frac{\mu_{\max}}{100}$. The tuning grid of Dirty models is given in sections:dirty. In practice we start by sampling 15 points on the base of the triangle that we further divide by a logarithmic sequence between $\lambda_{\max} = \|\boldsymbol{X}^\top Y\|_{2\infty}$ and $\frac{\lambda_{\max}}{100}$. Moreover, we sample 20 points over the line $y = \mu_{\max}$ for exclusive Group Lasso models. Figure C.1 shows an illustrative example of the sampled hyparaparameters.

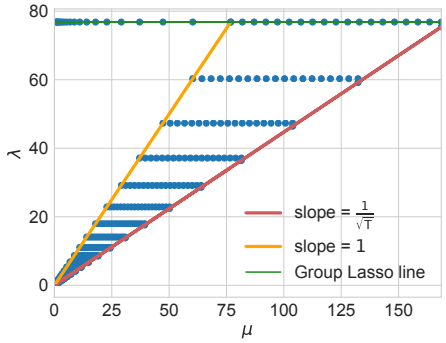


Figure C.1: Illustration of a hyperparamers grid sampling for Dirty models.

For MTW, μ is chosen among 10 candidates within a logarithmic scale between 1 and 100. The list of 20 values of λ is the same as the one used for the independent Lasso models.

MEG source localization The supports of the simulated brain activations (regression coefficients) are selected by taking one non-zero feature in each region illustrated in Figure C.2. If a regression coefficient is k-sparse, k regions are selected in which one random feature is non-zero.

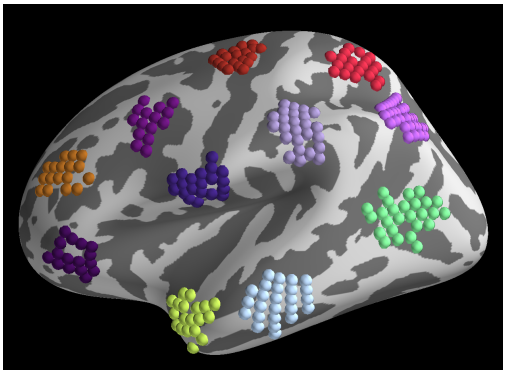
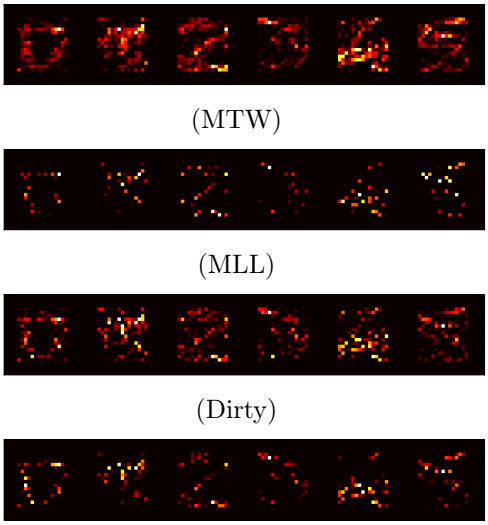


Figure C.2: Areas from which non-zero features are selected.

Handwritten digits recognition. We concatenate the handwritten digits dataset of van Breukelen et al. [1998] as a matrix $X \in \mathbb{R}^{nt \times p}$ where we selected the 6 first tasks (corresponding to the 6 first numerals 0-5) i.e $T = 6$; and the number of features $p = 240$ corresponding to 15×16 reduced images. The number of samples per task n is set to 10; 15; 20 and 50. We concatenate the one-hot encoded binary vector for each task $Y^t \in \mathbb{R}^{nT}$ so as to perforum one versus all classification. Thus, X is the design matrix common to all regression tasks. For each task, the dataset contains



(MTW)

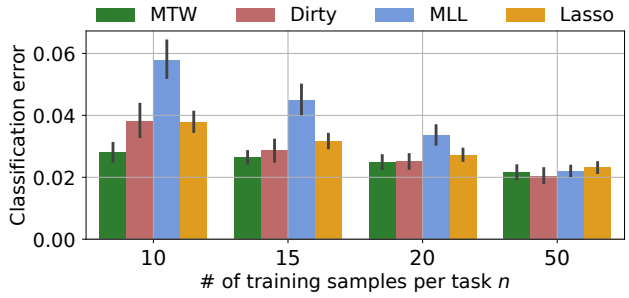
(MLL)

(Dirty)

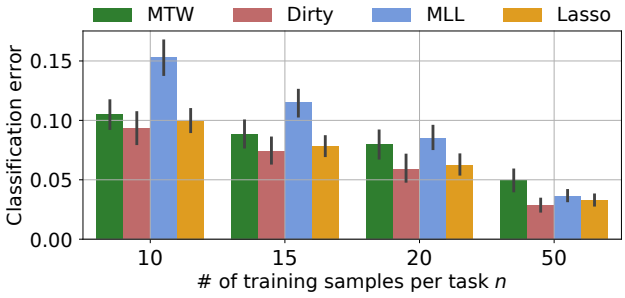
(Lasso)

Figure C.3: Learned regression coefficients $\boldsymbol{\theta}_+$ corresponding to the digits ('0'–'5').

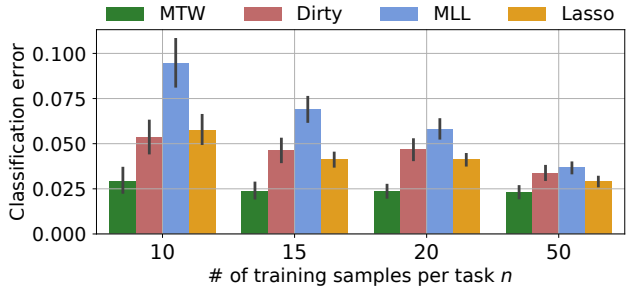
200 samples. Model selection if performed by first isolating a validation set of 200 - n samples per task. And computing a 5-fols Cross-validation error score on the training set. We performed 20 random selections of the validation samples and reported the mean classification errors in Figure 3. The detailed classification errors per task (taking the mean only across randomized splits) are displayed in Figure C.4. We display in Figure C.3 the learned regression coefficients by all methods.



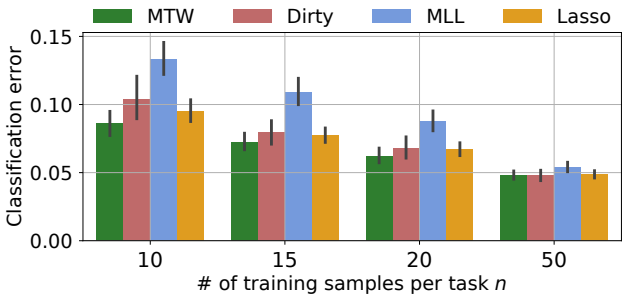
(Digit 0)



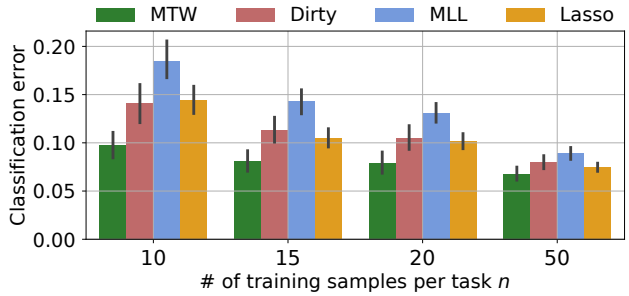
(Digit 1)



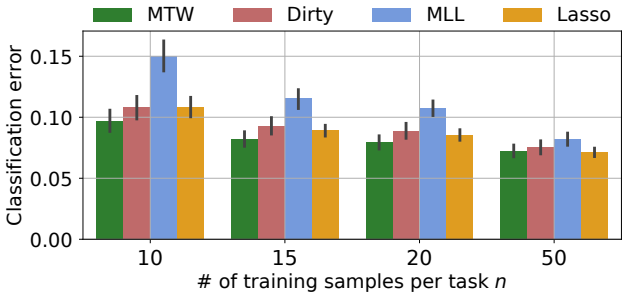
(Digit 2)



(Digit 3)



(Digit 4)



(Digit 5)

Figure C.4: Mean classification error per task (digit in ('0'–'5').)