Clustered Multi-Task Learning: a Convex Formulation

Laurent Jacob
Mines ParisTech – CBIO
INSERM U900, Institut Curie
35, rue Saint Honoré, 77300 Fontainebleau, France
laurent.jacob@mines-paristech.fr

Francis Bach
INRIA – Willow Project
Ecole Normale Supérieure,
45, rue d'Ulm, 75230 Paris, France
francis.bach@mines.org

Jean-Philippe Vert
Mines ParisTech – CBIO
INSERM U900, Institut Curie
35, rue Saint Honoré, 77300 Fontainebleau, France
jean-philippe.vert@mines-paristech.fr

Abstract

In multi-task learning several related tasks are considered simultaneously, with the hope that by an appropriate sharing of information across tasks, each task may benefit from the others. In the context of learning linear functions for supervised classification or regression, this can be achieved by including a priori information about the weight vectors associated with the tasks, and how they are expected to be related to each other. In this paper, we assume that tasks are clustered into groups, which are unknown beforehand, and that tasks within a group have similar weight vectors. We design a new spectral norm that encodes this a priori assumption, without the prior knowledge of the partition of tasks into groups, resulting in a new convex optimization formulation for multi-task learning. We show in simulations on synthetic examples and on the IEDB MHC-I binding dataset, that our approach outperforms well-known convex methods for multi-task learning, as well as related non-convex methods dedicated to the same problem.

1 Introduction

Regularization has emerged as a dominant theme in machine learning and statistics, providing an intuitive and principled tool for learning from high-dimensional data. In particular, regularization by squared Euclidean norms or squared Hilbert norms has been thoroughly studied in various settings, leading to efficient practical algorithms based on linear algebra, and to very good theoretical understanding (see, e.g., [1, 2]). In recent years, regularization by non Hilbert norms, such as ℓ^p norms with $p \neq 2$, has also generated considerable interest for the inference of linear functions in supervised classification or regression. Indeed, such norms can sometimes both make the problem statistically and numerically better-behaved, and impose various prior knowledge on the problem. For example, the ℓ^1-norm (the sum of absolute values) imposes some of the components to be equal to zero and is widely used to estimate sparse functions [3], while various combinations of ℓ^p norms can be defined to impose various sparsity patterns.

While most recent work has focused on studying the properties of simple well-known norms, we take the opposite approach in this paper. That is, assuming a given prior knowledge, how can we design a norm that will enforce it?

More precisely, we consider the problem of multi-task learning, which has recently emerged as a very promising research direction for various applications [4]. In multi-task learning several related inference tasks are considered simultaneously, with the hope that by an appropriate sharing

of information across tasks, each one may benefit from the others. When linear functions are estimated, each task is associated with a weight vector, and a common strategy to design multi-task learning algorithm is to translate some prior hypothesis about how the tasks are related to each other into constraints on the different weight vectors. For example, such constraints are typically that the weight vectors of the different tasks belong (a) to a Euclidean ball centered at the origin [5], which implies no sharing of information between tasks apart from the size of the different vectors, *i.e.*, the amount of regularization, (b) to a ball of unknown center [5], which enforces a similarity between the different weight vectors, or (c) to an unknown low-dimensional subspace [6, 7].

In this paper, we consider a different prior hypothesis that we believe could be more relevant in some applications: the hypothesis that *the different tasks are in fact clustered into different groups, and that the weight vectors of tasks within a group are similar to each other*. A key difference with [5], where a similar hypothesis is studied, is that we don't assume that the groups are known a priori, and in a sense our goal is both to identify the clusters and to use them for multi-task learning. An important situation that motivates this hypothesis is the case where most of the tasks are indeed related to each other, but a few "outlier" tasks are very different, in which case it may be better to impose similarity or low-dimensional constraints only to a subset of the tasks (thus forming a cluster) rather than to all tasks. Another situation of interest is when one can expect a natural organization of the tasks into clusters, such as when one wants to model the preferences of customers and believes that there are a few general types of customers with similar preferences within each type, although one does not know beforehand which customers belong to which types. Besides an improved performance if the hypothesis turns out to be correct, we also expect this approach to be able to identify the cluster structure among the tasks as a by-product of the inference step, e.g., to identify outliers or groups of customers, which can be of interest for further understanding of the structure of the problem.

In order to translate this hypothesis into a working algorithm, we follow the general strategy mentioned above which is to design a norm or a penalty over the set of weights which can be used as regularization in classical inference algorithms. We construct such a penalty by first assuming that the partition of the tasks into clusters is known, similarly to [5]. We then attempt to optimize the objective function of the inference algorithm over the set of partitions, a strategy that has proved useful in other contexts such as multiple kernel learning [8]. This optimization problem over the set of partitions being computationally challenging, we propose a convex relaxation of the problem which results in an efficient algorithm.

2 Multi-task learning with clustered tasks

We consider m related inference tasks that attempt to learn linear functions over $\mathcal{X} = \mathbb{R}^d$ from a training set of input/output pairs $(x_i, y_i)_{i=1,\dots,n}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. In the case of binary classification we usually take $\mathcal{Y} = \{-1, +1\}$, while in the case of regression we take $\mathcal{Y} = \mathbb{R}$. Each training example (x_i, y_i) is associated to a particular task $t \in [1, m]$, and we denote by $\mathcal{I}(t) \subset [1, n]$ the set of indices of training examples associated to the task t. Our goal is to infer m linear functions $f_t(x) = w_t^\top x$, for $t = 1, \dots, m$, associated to the different tasks. We denote by $W = (w_1 \dots w_m)$ the $d \times m$ matrix whose columns are the successive vectors we want to estimate.

We fix a loss function $l : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}$ that quantifies by $l(f(x), y)$ the cost of predicting $f(x)$ for the input x when the correct output is y. Typical loss functions include the square error in regression $l(u, y) = \frac{1}{2}(u - y)^2$ or the hinge loss in binary classification $l(u, y) = \max(0, 1 - uy)$ with $y \in \{-1, 1\}$. The empirical risk of a set of linear classifiers given in the matrix W is then defined as the average loss over the training set:

$$\ell(W) = \frac{1}{n} \sum_{t=1}^{m} \sum_{i \in \mathcal{I}(t)} l(w_t^\top x_i, y_i) \,. \tag{1}$$

In the sequel, we will often use the $m \times 1$ vector $\mathbf{1}$ composed of ones, the $m \times m$ projection matrices $U = \mathbf{1}\mathbf{1}^\top / m$ whose entries are all equal to $1/m$, as well as the projection matrix $\Pi = I - U$.

In order to learn simultaneously the m tasks, we follow the now well-established approach which looks for a set of weight vectors W that minimizes the empirical risk regularized by a penalty functional, *i.e.*, we consider the problem:

$$\min_{W \in \mathbb{R}^{d \times m}} \ell(W) + \lambda \Omega(W) \,, \tag{2}$$

where $\Omega(W)$ can be designed from prior knowledge to constrain some sharing of information between tasks. For example, [5] suggests to penalize both the norms of the w_i's and their variance,

i.e., to consider a function of the form:

$$\Omega_{variance}(W) = \|\bar{w}\|^2 + \frac{\beta}{m}\sum_{i=1}^{m}\|w_i - \bar{w}\|^2\,, \tag{3}$$

where $\bar{w} = \left(\sum_{i=1}^{n} w_i\right)/m$ is the mean weight vector. This penalty enforces a clustering of the $w_i's$ towards their mean when β increases. Alternatively, [7] propose to penalize the trace norm of W:

$$\Omega_{trace}(W) = \sum_{i=1}^{\min(d,m)} \sigma_i(W)\,, \tag{4}$$

where $\sigma_1(W), \ldots, \sigma_{\min(d,m)}(W)$ are the successive singular values of W. This enforces a low-rank solution in W, *i.e.*, constrains the different w_i's to live in a low-dimensional subspace.

Here we would like to define a penalty function $\Omega(W)$ that encodes as prior knowledge that tasks are clustered into $r < m$ groups. To do so, let us first assume that we know beforehand the clusters, *i.e.*, we have a partition of the set of tasks into r groups. In that case we can follow an approach proposed by [5] which for clarity we rephrase with our notations and slightly generalize now. For a given cluster $c \in [1, r]$, let us denote $\mathcal{J}(c) \subset [1, m]$ the set of tasks in c, $m_c = |\mathcal{J}(c)|$ the number of tasks in the cluster c, and E the $m \times r$ binary matrix which describes the cluster assignment for the m tasks, *i.e.*, $E_{ij} = 1$ if task i is in cluster j, 0 otherwise. Let us further denote by $\bar{w}_c = \left(\sum_{i\in\mathcal{J}(c)} w_i\right)/m_c$ the average weight vector for the tasks in c, and recall that $\bar{w} = \left(\sum_{i=1}^{m} w_i\right)/m$ denotes the average weight vector over all tasks. Finally it will be convenient to introduce the matrix $M = E(E^\top E)^{-1}E^\top$. M can also be written $I - L$, where L is the normalized Laplacian of the graph G whose nodes are the tasks connected by an edge if and only if they are in the same cluster. Then we can define three semi-norms of interest on W that quantify different orthogonal aspects:

- A global penalty, which measures on average how large the weight vectors are:

$$\Omega_{mean}(W) = n\|\bar{w}\|^2 = \mathrm{tr}WUW^\top\,.$$

- A measure of between-cluster variance, which quantifies how close to each other the different clusters are:

$$\Omega_{between}(W) = \sum_{c=1}^{r} m_c\|\bar{w}_c - \bar{w}\|^2 = \mathrm{tr}W(M-U)W^\top.$$

- A measure of within-cluster variance, which quantifies the compactness of the clusters:

$$\Omega_{within}(W) = \sum_{c=1}^{r}\left\{\sum_{i\in\mathcal{J}(c)}\|w_i - \bar{w}_c\|^2\right\} = \mathrm{tr}W(I-M)W^\top\,.$$

We note that both $\Omega_{between}(W)$ and $\Omega_{within}(W)$ depend on the particular choice of clusters E, or equivalently of M. We now propose to consider the following general penalty function:

$$\Omega(W) = \varepsilon_M\Omega_{mean}(W) + \varepsilon_B\Omega_{between}(W) + \varepsilon_W\Omega_{within}(W)\,, \tag{5}$$

where $\varepsilon_M, \varepsilon_B$ and ε_W are non-negative parameters that can balance the importance of the components of the penalty. Plugging this quadratic penalty into (2) leads to the general problem:

$$\min_{W\in\mathbb{R}^{d\times m}} \ell(W) + \lambda\mathrm{tr}W\Sigma(M)^{-1}W^\top\,, \tag{6}$$

where

$$\Sigma(M)^{-1} = \varepsilon_M U + \varepsilon_B(M-U) + \varepsilon_W(I-M)\,. \tag{7}$$

Here we use the notation $\Sigma(M)$ to insist on the fact that this quadratic penalty depends on the cluster structure through the matrix M. Observing that the matrices U, $M-U$ and $I-M$ are orthogonal projections onto orthogonal supplementary subspaces, we easily get from (7):

$$\Sigma(M) = \varepsilon_M^{-1}U + \varepsilon_B^{-1}(M-U) + \varepsilon_W^{-1}(I-M) = \varepsilon_W^{-1}I + (\varepsilon_M^{-1} - \varepsilon_B^{-1})U + (\varepsilon_B^{-1} - \varepsilon_W^{-1})M\,. \tag{8}$$

By choosing particular values for $\varepsilon_M, \varepsilon_B$ and ε_W we can recover several situations, In particular:

- For $\varepsilon_W = \varepsilon_B = \varepsilon_M = \varepsilon$, we simply recover the Frobenius norm of W, which does not put any constraint on the relationship between the different tasks:

$$\Omega(W) = \varepsilon\mathrm{tr}WW^\top = \varepsilon\sum_{i=1}^{m}\|w_i\|^2\,.$$

- For $\varepsilon_W = \varepsilon_B > \varepsilon_M$, we recover the penalty of [5] without clusters:
$$\Omega(W) = \mathrm{tr}W\left(\varepsilon_M U + \varepsilon_B(I-U)\right)W^\top = \varepsilon_M n\|\bar{w}\|^2 + \varepsilon_B \sum_{i=1}^m \|w_i - \bar{w}\|^2 .$$
In that case, a global similarity between tasks is enforced, in addition to the general constraint on their mean. The structure in clusters plays no role since the sum of the between- and within-cluster variance is independent of the particular choice of clusters.
- For $\varepsilon_W > \varepsilon_B = \varepsilon_M$ we recover the penalty of [5] with clusters:

$$\Omega(W) = \mathrm{tr}W\left(\varepsilon_M M + \varepsilon_W(I-M)\right)W^\top = \varepsilon_M \sum_{c=1}^r \left\{ m_c\|\bar{w}_c\|^2 + \tfrac{\varepsilon_W}{\varepsilon_M}\sum_{i\in\mathcal{J}(c)}\|w_i - \bar{w}_c\|^2 \right\} .$$

In order to enforce a cluster hypothesis on the tasks, we therefore see that a natural choice is to take $\varepsilon_W > \varepsilon_B > \varepsilon_M$ in (5). This would have the effect of penalizing more the within-cluster variance than the between-cluster variance, hence promoting compact clusters. Of course, a major limitation at this point is that we assumed the cluster structure known a priori (through the matrix E, or equivalently M). In many cases of interest, we would like instead to learn the cluster structure itself from the data. We propose to learn the cluster structure in our framework by optimizing our objective function (6) both in W and M, *i.e.*, to consider the problem:

$$\min_{W\in\mathbb{R}^{d\times m}, M\in\mathcal{M}_r} \ell(W) + \lambda \mathrm{tr}W\Sigma(M)^{-1}W^\top , \tag{9}$$

where $\mathcal{M}_r$ denotes the set of matrices $M = E(E^\top E)^{-1}E^\top$ defined by a clustering of the m tasks into r clusters and $\Sigma(M)$ is defined in (8). Denoting by $\mathcal{S}_r = \{\Sigma(M) : M \in \mathcal{M}_r\}$ the corresponding set of positive semidefinite matrices, we can equivalently rewrite the problem as:

$$\min_{W\in\mathbb{R}^{d\times m}, \Sigma\in\mathcal{S}_r} \ell(W) + \lambda \mathrm{tr}W\Sigma^{-1}W^\top . \tag{10}$$

The objective function in (10) is jointly convex in $W \in \mathbb{R}^{d\times m}$ and $\Sigma \in \mathcal{S}_+^m$, the set of $m\times m$ positive semidefinite matrices, however the (finite) set $\mathcal{S}_r$ is not convex, making this problem intractable. We are now going to propose a convex relaxation of (10) by optimizing over a convex set of positive semidefinite matrices that contains $\mathcal{S}_r$.

3 Convex relaxation

In order to formulate a convex relaxation of (10), we observe that in the penalty term (5) the cluster structure only contributes to the second and third terms $\Omega_{between}(W)$ and $\Omega_{within}(W)$, and that these penalties only depend on the centered version of W. In terms of matrices, only the last two terms of $\Sigma(M)^{-1}$ in (7) depend on M, *i.e.*, on the clustering, and these terms can be re-written as:

$$\varepsilon_B(M-U) + \varepsilon_W(I-M) = \Pi(\varepsilon_B M + \varepsilon_W(I-M))\Pi. \tag{11}$$

Indeed, it is easy to check that $M - U = M\Pi = \Pi M\Pi$, and that $I - M = I - U - (M-U) = \Pi - \Pi M\Pi = \Pi(I-M)\Pi$. Intuitively, multiplying by Π on the right (*resp.* on the left) centers the rows (*resp.* the columns) of a matrix, and both $M-U$ and $I-M$ are row- and column-centered.

To simplify notations, let us introduce $\widetilde{M} = \Pi M\Pi$. Plugging (11) in (7) and (9), we get the penalty

$$\mathrm{tr}W\Sigma(M)^{-1}W^\top = \varepsilon_M\left(\mathrm{tr}W^\top WU\right) + (W\Pi)(\varepsilon_B\widetilde{M} + \varepsilon_W(I-\widetilde{M}))(W\Pi)^\top, \tag{12}$$

in which, again, only the second part needs to be optimized with respect to the clustering M. Denoting $\Sigma_c^{-1}(M) = \varepsilon_B\widetilde{M} + \varepsilon_W(I-\widetilde{M})$, one can express $\Sigma_c(M)$, using the fact that $\widetilde{M}$ is a projection:

$$\Sigma_c(M) = \left(\varepsilon_B^{-1} - \varepsilon_W^{-1}\right)\widetilde{M} + \varepsilon_W^{-1} I. \tag{13}$$

Σ_c is characterized by $\widetilde{M} = \Pi M\Pi$, that is discrete by construction, hence the non-convexity of $\mathcal{S}_r$. We have the natural constraints $M \succeq 0$ (*i.e.*, $\widetilde{M} \succeq -U$), $0 \preceq M \preceq I$ (*i.e.*, $0 \preceq \widetilde{M} \preceq \Pi$) and $\mathrm{tr}M = r$ (*i.e.*, $\mathrm{tr}\widetilde{M} = r-1$). A possible convex relaxation of the discrete set of matrices $\widetilde{M}$ is therefore $\{\widetilde{M} : 0 \preceq \widetilde{M} \preceq I,\ \mathrm{tr}\widetilde{M} = r-1\}$. This gives an equivalent convex set $\mathcal{S}_c$ for Σ_c, namely:

$$\mathcal{S}_c = \left\{\Sigma_c \in \mathcal{S}_+^m : \alpha I \preceq \Sigma_c \preceq \beta I, \mathrm{tr}\Sigma_c = \gamma\right\} , \tag{14}$$

with $\alpha = \varepsilon_W^{-1}$, $\beta = \varepsilon_B^{-1}$ and $\gamma = (m-r+1)\varepsilon_W^{-1} + (r-1)\varepsilon_B^{-1}$. Incorporating the first part of the penalty (12) into the empirical risk term by defining $\ell_c(W) = \lambda\ell(W) + \varepsilon_M\left(\mathrm{tr}W^\top WU\right)$, we are now ready to state our relaxation of (10):

$$\min_{W\in\mathbb{R}^{d\times m}, \Sigma_c\in\mathcal{S}_c} \ell_c(W) + \lambda \mathrm{tr}W\Pi\Sigma_c^{-1}(W\Pi)^\top . \tag{15}$$

3.1 Reinterpretation in terms of norms

We denote $\|W\|_c^2 = \min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W \Sigma_c^{-1} W^T$ the *cluster norm* (CN). For any convex set $\mathcal{S}_c$, we obtain a norm on W (that we apply here to its centered version). By putting some different constraints on the set $\mathcal{S}_c$, we obtain different norms on W, and in fact all previous multi-task formulations may be cast in this way, *i.e.*, by choosing a specific set of positive matrices $\mathcal{S}_c$ (*e.g.*, trace constraint for the trace norm, and simply a singleton for the Frobenius norm). Thus, designing norms for multi-task learning is equivalent to designing a set of positive matrices. In this paper, we have investigated a specific set adapted for clustered-tasks, but other sets could be designed in other situations.

Note that we have selected a simple *spectral* convex set $\mathcal{S}_c$ in order to make the optimization simpler in Section 3.3, but we could also add some additional constraints that encode the point-wise positivity of the matrix M. Finally, when $r = 1$ (one cluster) and $r = m$ (one cluster per task), we get back the formulation of [5].

3.2 Reinterpretation as a convex relaxation of K-means

In this section we show that the semi-norm $\|W\Pi\|_c^2$ that we have designed earlier, can be interpreted as a convex relaxation of K-means on the tasks [9]. Indeed, given $W \in \mathbb{R}^{d \times m}$, K-means aims to decompose it in the form $W = \mu E^\top$ where $\mu \in \mathbb{R}^{d \times r}$ are cluster centers and E represents a partition. Given E, μ is found by minimizing $\min_\mu \|W^\top - E\mu^\top\|_F^2$. Thus, a natural strategy outlined by [9], is to alternate between optimizing μ, the partition E and the weight vectors W. We now show that our convex norm is obtained when minimizing in closed form with respect to μ and relaxing.

By translation invariance, this is equivalent to minimizing $\min_\mu \|\Pi W^\top - \Pi E \mu^\top\|_F^2$. If we add a penalization on μ of the form $\lambda \mathrm{tr} E^\top E \mu \mu^\top$, then a short calculation shows that the minimum with respect to μ (*i.e.*, after optimization of the cluster centers) is equal to

$$\mathrm{tr}\Pi W^\top W \Pi (\Pi E (E^\top E)^{-1} E^\top \Pi/\lambda + I)^{-1} = \mathrm{tr}\Pi W^\top W \Pi (\Pi M \Pi/\lambda + I)^{-1}.$$

By comparing with Eq. (13), we see that our formulation is indeed a convex relaxation of K-means.

3.3 Primal optimization

Let us now show in more details how (15) can be solved efficiently. Whereas a dual formulation could be easily derived following [8], a direct approach is to rewrite (15) as

$$\min_{W \in \mathbb{R}^{d \times m}} \left(\ell_c(W) + \min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W \Pi \Sigma_c^{-1} (W\Pi)^\top \right) \tag{16}$$

which, if ℓ_c is differentiable, can be directly optimized by gradient-based methods on W since $\|W\Pi\|_c^2 = \min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W \Pi \Sigma_c^{-1} (W\Pi)^\top$ is a quadratic semi-norm of $W\Pi$. This regularization term $\mathrm{tr} W \Pi \Sigma_c^{-1} (W\Pi)^\top$ can be computed efficiently using a semi-closed form. Indeed, since Σ_c as defined in (14) is a spectral set (*i.e.*, it does depend only on eigenvalues of covariance matrices), we obtain a function of the singular values of $W\Pi$ (or equivalently the eigenvalues of $W\Pi W^\top$):

$$\min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W \Pi \Sigma_c^{-1} (W\Pi)^\top = \min_{\lambda \in \mathbb{R}^m,\, \alpha \leq \lambda_i \leq \beta,\, \lambda \mathbf{1} = \gamma,\, V \in \mathcal{O}^m} \mathrm{tr} W \Pi V \,\mathrm{diag}(\lambda)^{-1} V^\top (W\Pi)^\top,$$

where $\mathcal{O}^m$ is the set of orthogonal matrices in $\mathbb{R}^{m \times m}$. The optimal V is the matrix of the eigenvectors of $W\Pi W^\top$, and we obtain the value of the objective function at the optimum:

$$\min_{\Sigma \in S} \mathrm{tr} W \Pi \Sigma^{-1} (W\Pi)^\top = \min_{\lambda \in \mathbb{R}^m,\, \alpha \leq \lambda_i \leq \beta,\, \lambda \mathbf{1} = \gamma} \sum_{i=1}^m \frac{\sigma_i^2}{\lambda_i},$$

where σ and λ are the vectors containing the singular values of $W\Pi$ and Σ respectively. Now, we simply need to be able to compute this function of the singular values.

The only coupling in this formulation comes from the trace constraint. The Lagrangian corresponding to this constraint is:

$$\mathcal{L}(\lambda, \nu) = \sum_{i=1}^m \frac{\sigma_i^2}{\lambda_i} + \nu \left(\sum_{i=1}^m \lambda_i - \gamma \right). \tag{17}$$

For $\nu \leq 0$, this is a decreasing function of λ_i, so the minimum on $\lambda_i \in [\alpha, \beta]$ is reached for $\lambda_i = \beta$. The dual function is then a linear non-decreasing function of ν (since $\alpha \leq \gamma/m \leq \beta$ from the definition of α, β, γ in (14)), which reaches it maximum value (on $\nu \leq 0$) at $\nu = 0$. Let us therefore now consider the dual for $\nu \geq 0$. (17) is then a convex function of λ_i. Canceling its derivative with respect to λ_i gives that the minimum in $\lambda \in \mathbb{R}$ is reached for $\lambda_i = \sigma_i / \sqrt{\nu}$. Now this may not be

in the constraint set (α, β), so if $\sigma_i < \alpha\sqrt{\nu}$ then the minimum in $\lambda_i \in [\alpha, \beta]$ of (17) is reached for $\lambda_i = \alpha$, and if $\sigma_i > \beta\sqrt{\nu}$ it is reached for $\lambda_i = \beta$. Otherwise, it is reached for $\lambda_i = \sigma_i/\sqrt{\nu}$. Reporting this in (17), the dual problem is therefore

$$\max_{\nu \geq 0} \sum_{i, \alpha\sqrt{\nu} \leq \sigma_i \leq \beta\sqrt{\nu}} 2\sigma_i\sqrt{\nu} + \sum_{i, \sigma_i < \alpha\sqrt{\nu}} \left(\frac{\sigma_i^2}{\alpha} + \nu\alpha\right) + \sum_{i, \beta\sqrt{\nu} < \sigma_i} \left(\frac{\sigma_i^2}{\beta} + \nu\beta\right) - \nu\gamma \,. \quad (18)$$

Since a closed form for this expression is known for each fixed value of ν, one can obtain $\|W\Pi\|_c^2$ (and the eigenvalues of Σ^*) by Algorithm 1. The cancellation condition in Algorithm 1 is that the

Algorithm 1 Computing $\|A\|_c^2$

Require: A, α, β, γ.
Ensure: $\|A\|_c^2$, λ^*.
  Compute the singular values σ_i of A.
  Order the $\frac{\sigma_i^2}{\alpha^2}, \frac{\sigma_i^2}{\beta^2}$ in a vector I (with an additional 0 at the beginning).
  for all interval (a, b) of I **do**
    if $\frac{\partial \mathcal{L}(\lambda^*, \nu)}{\partial \nu}$ is canceled on $\nu \in (a, b)$ **then**
      Replace ν^* in the dual function $\mathcal{L}(\lambda^*, \nu)$ to get $\|A\|_c^2$, compute λ^* on (a, b).
      return $\|A\|_c^2$, λ^*.
    end if
  end for

value canceling the derivative belongs to (a, b), *i.e.*,

$$\nu = \left(\frac{\sum_{i, \alpha\sqrt{\nu} \leq \sigma_i \leq \beta\sqrt{\nu}} \sigma_i}{\gamma - (\alpha n^- + \beta n^+)}\right)^2 \in (a, b) \,,$$

where n^- and n^+ are the number of $\sigma_i < \alpha\sqrt{\nu}$ and $\sigma_i > \beta\sqrt{\nu}$ respectively. Denoting $\|A\|_c^2 = F(A, \Sigma^*(A))$, $\nabla_A F = \partial_A F + \partial_\Sigma F \partial_A \Sigma$ cannot be computed because of the non-differentiable constraints on Σ for F. We followed an alternative direction, using only the $\partial_A F$ part.

4 Experiments

4.1 Artificial data

We generated synthetic data consisting of two clusters of two tasks. The tasks are vectors of $\mathbb{R}^d$, $d = 30$. For each cluster, a center $\bar{w}_c$ was generated in $\mathbb{R}^{d-2}$, so that the two clusters be orthogonal. More precisely, each $\bar{w}_c$ had $(d-2)/2$ random features randomly drawn from $\mathcal{N}(0, \sigma_r^2)$, $\sigma_r^2 = 900$, and $(d-2)/2$ zero features. Then, each tasks t was computed as $w_t + \bar{w}_c(t)$, where $c(t)$ was the cluster of t. w_t had the same zero feature as its cluster center, and the other features were drawn from $\mathcal{N}(0, \sigma_c^2)$, $\sigma_c^2 = 16$. The last two features were non-zero for all the tasks and drawn from $\mathcal{N}(0, \sigma_c^2)$. For each task, 2000 points were generated and a normal noise of variance $\sigma_n^2 = 150$ was added.

In a first experiment, we compared our cluster norm $\|.\|_c^2$ with the single-task learning given by the Frobenius norm, and with the trace norm, that corresponds to the assumption that the tasks live in a low-dimension space. The multi-task kernel approach being a special case of CN, its performance will always be between the performance of the single task and the performance of CN.

In a second setting, we compare CN to alternative methods that differ in the way they learn Σ:

- The *True metric* approach, that simply plugs the actual clustering in E and optimizes W using this fixed metric. This necessitates to know the true clustering *a priori*, and can be thought of like a golden standard.
- The *k-means* approach, that alternates between optimizing the tasks in W given the metric Σ and re-learning Σ by clustering the tasks w_i [9]. The clustering is done by a k-means run 3 times. This is a non convex approach, and different initialization of k-means may result in different local minima.

We also tried one run of CN followed by a run of *True metric* using the learned Σ reprojected in $\mathcal{S}_r$ by rounding, *i.e.*, by performing k-means on the eigenvectors of the learned Σ (*Reprojected* approach), and a run of *k-means* starting from the relaxed solution (*CNinit* approach).

Only the first method requires to know the true clustering a priori, all the other methods can be run without any knowledge of the clustering structure of the tasks.

Each method was run with different numbers of training points. The training points were equally separated between the two clusters and for each cluster, $5/6$th of the points were used for the first task and $1/6$th for the second, in order to simulate a natural setting were some tasks have fewer data. We used the 2000 points of each task to build 3 training folds, and the remaining points were used for testing. We used the mean RMSE across the tasks as a criterion, and a quadratic loss for $\ell(W)$.

The results of the first experiment are shown on Figure 1 (left). As expected, both multi-task approaches perform better than the approach that learns each task independently. CN penalization on the other hand always gives better testing error than the trace norm penalization, with a stronger advantage when very few training points are available. When more training points become available, all the methods give more and more similar performances. In particular, with large samples, it is not useful anymore to use a multi-task approach.

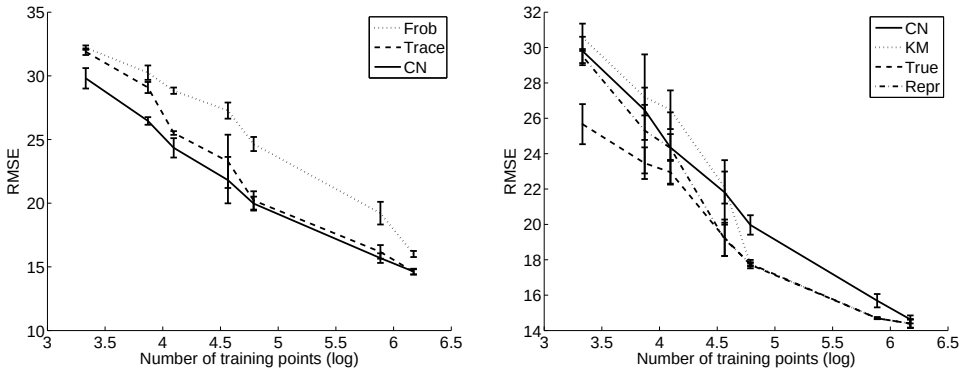

Figure 1: RMSE versus number of training points for the tested methods.

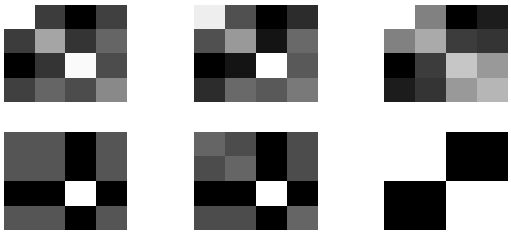

Figure 2: Recovered Σ with CN (upper line) and k-means (lower line) for 28, 50 and 100 points.

Figure 1 (right) shows the results of the second experiment. Using the true metric always gives the best results. For 28 training points, no method recovers the correct clustering structure, as displayed on Figure 2, although CN performs slightly better than the *k-means* approach since the metric it learns is more diffuse. For 50 training points, CN performs much better than the *k-means* approach, which completely fails to recover the clustering structure as illustrated by the Σ learned for 28 and 50 training points on Figure 2. In the latter setting, CN partially recovers the clusters. When more training points become available, the *k-means* approach perfectly recovers the clustering structure and outperforms the relaxed approach. The reprojected approach, on the other hand, performs always as well as the best of the two other methods. The CNinit approach results are not displayed since the are the same as for the reprojected method.

4.2 MHC-I binding data

We also applied our method to the IEDB MHC-I peptide binding benchmark proposed in [10]. This database contains binding affinities of various peptides, *i.e.*, short amino-acid sequences, with different MHC-I molecules. This binding process is central in the immune system, and predicting it is crucial, for example to design vaccines. The affinities are thresholded to give a prediction problem. Each MHC-I molecule is considered as a task, and the goal is to predict whether a peptide binds a molecule. We used an orthogonal coding of the amino acids to represent the peptides and balanced

Table 1: Prediction error for the 10 molecules with less than 200 training peptides in IEDB.

Method	Pooling	Frobenius norm	Multi-task kernel	Trace norm	Cluster norm
Test error	$26.53\% \pm 2.0$	$11.62\% \pm 1.4$	$10.10\% \pm 1.4$	$9.20\% \pm 1.3$	$8.71\% \pm 1.5$

the data by keeping only one negative example for each positive point, resulting in 15236 points involving 35 different molecules. We chose a logistic loss for $\ell(W)$.

Multi-task learning approaches have already proved useful for this problem, see for example [11, 12]. Besides, it is well known in the vaccine design community that some molecules can be grouped into empirically defined *supertypes* known to have similar binding behaviors.

[12] showed in particular that the multi-task approaches were very useful for molecules with few known binders. Following this observation, we consider the mean error on the 10 molecules with less than 200 known ligands, and report the results in Table 1. We did not select the parameters by internal cross validation, but chose them among a small set of values in order to avoid overfitting. More accurate results could arise from such a cross validation, in particular concerning the number of clusters (here we limited the choice to 2 or 10 clusters).

The pooling approach simply considers one global prediction problem by pooling together the data available for all molecules. The results illustrate that it is better to consider individual models than one unique pooled model.On the other hand, all the multitask approaches improve the accuracy, the cluster norm giving the best performance. The learned Σ, however, did not recover the known supertypes, although it may contain some relevant information on the binding behavior of the molecules.

5 Conclusion

We have presented a convex approach to clustered multi-task learning, based on the design of a dedicated norm. Promising results were presented on synthetic examples and on the IEDB dataset. We are currently investigating more refined convex relaxations and the natural extension to non-linear multi-task learning as well as the inclusion of specific features on the tasks, which has shown to improve performance in other settings [6].

References

[1] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[2] F. Girosi, M. Jones, and T. Poggio. Regularization Theory and Neural Networks Architectures. *Neural Comput.*, 7(2):219–269, 1995.

[3] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Stat. Soc. B.*, 58:267–288, 1996.

[4] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003.

[5] T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.

[6] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. Low-rank matrix factorization with attributes. Technical Report cs/0611124, arXiv, 2006.

[7] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Adv. NIPS 19*, pages 41–48, Cambridge, MA, 2007. MIT Press.

[8] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[9] M. Deodhar and J. Ghosh. A framework for simultaneous co-clustering and learning from complex data. In *KDD '07*, pages 250–259, New York, NY, USA, 2007. ACM.

[10] B. Peters, H.-H Bui, S. Frankild, M. Nielson, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette. A community resource benchmarking predictions of peptide binding to MHC-I molecules. *PLoS Comput Biol*, 2(6):e65, 2006.

[11] D. Heckerman, D. Kadie, and J. Listgarten. Leveraging information across HLA alleles/supertypes improves epitope prediction. *J. Comput. Biol.*, 14(6):736–746, 2007.

[12] L. Jacob and J.-P. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, Feb 2008.