Sélection de variables pour l'apprentissage simultanée de tâches

Rémi Flamary, Alain Rakotomamonjy, Gilles Gasso, Stephane Canu

Sélection de variables pour l'apprentissage simultanée de tâches

R. Flamary, A. Rakotomamonjy , G. Gasso, S. Canu

LITIS EA 4108, INSA-Université de Rouen
76800 Saint Etienne du Rouvray, France

Résumé : Cette article traite de la sélection de variables pour l'apprentissage simultanée de taches de discrimination SVM . Nous formulons ce problème comme étant un apprentissage multi-taches avec pour terme de régularisation une norme mixte de type $\ell_p - \ell_2$ avec $p \leq 1$. Cette dernière permet d'obtenir des modèles de discrimination pour chaque tâche, utilisant un même sous-ensemble des variables. Nous proposons tout d'abord un algorithme permettant de résoudre le problème d'apprentissage lorsque la norme mixte est convexe ($p = 1$). Ensuite, à l'aide de la programmation DC, nous traitons le cas non-convexe ($p < 1$) . Nous montrons que ce dernier cas peut être résolu par un algorithme itératif où, à chaque itération, un problème basé sur la norme mixte $\ell_1 - \ell_2$ est résolu. Nos expériences montrent l'interêt de la méthode sur quelques problèmes de discriminations simultanées.

Mots-clés : Apprentissage multi-tâches, Sélection de variables, méthodes à noyaux

1 Introduction

Multi-Task Learning (MTL) is a statistical learning framework which seeks at learning different models in a joint manner. The idea behind this paradigm is that, when the tasks to be learned are similar enough or are related in some sense, it may be advantageous to take into account these relations between tasks. Several works have experimentally highlighted the benefit of such a framework (Caruana, 1997). However, the notion of relatedness between tasks is rather vague and depends on the problem at hand. For instance, one can consider that models resulting from related tasks should have similar norms (Evgeniou & Pontil, 2004; Kato *et al.*, 2008). In other works, task relatedness is represented through a probabilistic model (Yu *et al.*, 2005). Prior knowledge on tasks are then translated into an appropriated regularization term or into a hierarchical Bayesian model that can be handled by a learning algorithm.

In this work, we consider that tasks to be learned share a common subset of features or kernel representation. This means that while learning the tasks, we jointly look for features or kernels that are useful for all tasks. In this context of joint feature selection, for multiple related tasks several works have already been carried out. For instance, Jebara (2004) has introduced a maximum entropy discrimination for solving such a problem.

Some other works cast the problem into a probabilistic framework which uses automatic relevance determination and a hierarchical Bayesian model for selecting the relevant features (Bi *et al.*, 2008; Xiong *et al.*, 2006). Another trend considers to use a regularization principle and thus minimizes a regularized empirical risk while the regularization term favors a common sparsity profile in features for all tasks. Such an approach have been investigated by Argyriou *et al.* (2008) and Obozinski *et al.* (2007). In this latter work, the authors propose a $\ell_1 - \ell_2$ regularization term which can be interpreted as a convex extension of the sparsity-inducing ℓ_1 norm in single task learning.

This paper also considers this regularization principle for joint feature selection across tasks. Our contribution is two fold. First we consider the multi-task learning problem in a SVM framework with a kernel representation. The proposed algorithms rely on sparsity-inducing ($\ell_p - \ell_2$) mixed-norms regularizers which encourage sparse kernel selection among a prescribed set of kernels. This set of *basis* kernels can be made large enough at will, gathering information about the different sources of the input samples. From this framework, we show that our formulation in the convex case turns into a multiple kernel learning problem. Therefore, an efficient algorithm is derived based on off-the-shelf MKL solvers (Rakotomamonjy *et al.*, 2008). At the second stage, we extend the analysis to a non-convex regularization term in order to gain in sparsity The difficulty raised by this formulation is tackled via a DC programming (Horst & Thoai, 1999). This leads to an iterative scheme which solves at each iteration, a reweighted MTL problem.

In the next section, we present the general formulation of the sparse MTL problem as well as a brief review of closely spirit-related works. Algorithmic developments are presented in Section 3. Then, some empirical results that illustrate the behavior of our algorithms are given in Section 4 while some concluding remarks are drawn in Section 5.

2 Multi-Task feature/kernel selection framework

This section introduces our framework for sparse MTL and discusses the connection with other works.

2.1 Framework

Suppose we are given T classification tasks to be achieved from T different datasets $\{x_{i,1}, y_{i,1}\}_i^{n_1}, \cdots, \{x_{i,T}, y_{i,T}\}_i^{n_T}$, where any $x_{i,\cdot} \in \mathcal{X}$ and $y_{i,\cdot} \in \{+1, -1\}$ and n_i denotes the i^{th} dataset size. For a given task t, we are looking for a decision function of the form :

$$f_t(x) = \sum_{k=1}^{M} f_{t,k}(x) + b_t \qquad \forall t \in \{1, \cdots, T\} \tag{1}$$

where any function $f_{\cdot,k}$ belongs to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ of kernel K_k, b_t is the bias term and M is the number of *basis* kernels provided. Note that depending on the input space $\mathcal{X}$, $\mathcal{H}_k$ can be of different forms. For instance, if $\mathcal{X} = \mathbb{R}^d$,

$\mathcal{H}_k$ can be a subset based on a single or several dimensions of $\mathbb{R}^d$. $\mathcal{H}_k$ can be also an infinite dimension space.

The objective of this work is to learn the decision function f_t of each task under the constraints that all these functions share a common sparse profile of their kernel representation. Hence, the pursued hope is to build a learning algorithm able to yield many vanishing functions $f_{t,k}$ for all t.

For achieving this goal, we cast our problem as the following optimization problem :

$$\min_{f_1,\cdots,f_T} C \cdot \sum_{t,i} L(f_t(x_{i,t}), y_{i,t}) + \Omega(f_1, \cdots, f_T)$$

where $L(f_t(x), y)$ is a loss function, Ω a sparsity-inducing penalty function involving all f_t and C a trade-off parameter that balances both antagonist objectives.

2.2 Joint sparsity-inducing penalty

For a single task empirical minimization problem, sparse models are usually induced by the use of a ℓ_1-norm regularizer (Tibshirani, 1995). For a Multi-Task Learning problem, this approach can be properly generalized by the use of appropriate norm. For instance, Obozinski *et al.* (2007) propose a regularizer of the form

$$\Omega(f_1, \cdots, f_T) = \sum_{k=1}^{M} \left(\sum_{t=1}^{T} \|f_{t,k}\|^2_{\mathcal{H}_k} \right)^{1/2}$$

which is equivalent to a ℓ_1-norm for a single linear task. This regularizer can be generalized by

$$\Omega_{p,q}(f_1, \cdots, f_T) = \sum_{k=1}^{M} \left(\sum_{t=1}^{T} \|f_{t,k}\|^q_{\mathcal{H}_k} \right)^{p/q} \tag{2}$$

where typically $p \leq 1$ and $q \geq 1$. For this regularizer, a ℓ_q norm is applied to the vector of all task norms in $\mathcal{H}_k$ and then a ℓ_p pseudo-norm is applied to the resulting vector. The ℓ_q norm in the regularizer controls the weights of each task for the space $\mathcal{H}_k$ and how this kernel representation will be shared across tasks. For instance, large value of q (like $q = \infty$) means that as soon as $\|f_{t,k}\|_{\mathcal{H}_k}$ is non-zero, another task t' can have a non-zero norm for $f_{t',k}$ without increasing significantly the regularizer $\Omega_{p,q}$. The ℓ_p pseudo-norm controls the sparsity of the kernel representation for all tasks.

Such a regularizer has already been proposed for single task learning for achieving composite absolute penalization (Zhao *et al.*, to appear) or for composite kernel learning (Szafranski *et al.*, 2008).

However, in the context of multi-task learning, some particular cases of the mixed-norm $\Omega_{p,q}$ have been considered. Obozinski *et al.* (2007) use $p = 1$ and $q = 2$ while Liu *et al.* (2009) and Quattoni *et al.* (2008) propose the use of $p = 1$ and $q = \infty$. For all these works, the authors have focused on convex situations since $\Omega_{p,q}$ is known to be convex whenever $p, q \geq 1$ and non-convex for $p < 1$ and $q \geq 1$.

Recently, several works on sparse single learning models have stressed the need of non-convex penalties for achieving better sparsity. For instance, Knight & Fu (2000)

suggested the use of the so-called Bridge penalty which simply consists in replacing the ℓ_1 norm with a ℓ_p norm with $p < 1$. In our multi-task learning framework, this can be naturally generalized by using the regularizer given in Equation (2) with $p < 1$.

2.3 Relation with other works

Before delving into the details of algorithms for solving the $\ell_p - \ell_2$ regularized sparse MTL, let us relate the proposed approach to the recent similar methods.

As far as we know, the first works which proposed mixed-norms for joint-sparsity inducing regularizer come from the signal processing community (Cotter *et al.*, 2005; Tropp, 2006). These works have investigated the use of $\ell_1 - \ell_2$ and $\ell_1 - \ell_\infty$ penalties together with least-squares loss function for sparse signal approximations.

Due to its convexity, the penalty $\ell_1 - \ell_2$ has attracted many interests for jointly sparse multi-task learning. Indeed the seminal works of Argyriou *et al.* (2008) and Obozinski *et al.* (2007) have opened the road to regularized sparse MTL. These two works differ in their algorithmic approach : while Argyriou et al. proposed an alternating minimization approach, Obozinski et al. used an homotopy method for solving the problem. In both cases, their algorithms consider a smooth loss function. Several probabilistic approaches actually boil down to be equivalent to the use of $\ell_1 - \ell_2$ penalty (Xiong *et al.*, 2006; Bi *et al.*, 2008) and thus they simply provided a probabilistic interpretation of the work of Obozinski et al. More recently, Liu *et al.* (2009) and Quattoni *et al.* (2008) considered solving the multi-task learning by using a $\ell_1 - \ell_\infty$ regularization. While Liu et al. provided algorithms for smooth loss functions, Quattoni et al. considered the Hinge loss and derived a linear programming method for solving the resulting problem.

Our work differs from the previously mentioned in several ways. At first, we consider a kernel selection framework which is general enough to include feature selection or grouped-feature selection as a special case (Bach, 2008). Then, instead of considering smooth and differentiable loss functions, we use a hinge loss cost function as Quattoni *et al.* (2008). However, in the latter work, the features are extracted from an unsupervised KPCA projection of the labeled data onto the space spanned by some available unlabeled samples followed by a multi-task learning with a linear SVM and the mentioned $\ell_1 - \ell_\infty$ penalty.

According to us, the approach proposed hereafter is more general and explores the possibility of combining many different kernels. Furthermore, we benefit from the efficiency of SVM algorithm and multiple kernel learning tools upon which we built our convex MTL solver. Finally, we go beyond the convex cases and consider the use of a larger class of sparsity-inducing regularisation term which includes the $\ell_1 - \ell_2$ norm as a special case. The DC procedure allows to solve the problem as an iterative reweighted MTL problem.

3 Algorithms for jointly sparse multi-task SVM

In this section, we propose some algorithms for solving the sparse multi-task SVM problem when using $\Omega_{p,q}$ as a regularizer with $p \leq 1$ and $q = 2$. At first, we consider

the convex problem with $p = 1$ and then we introduce an algorithm which solves the problem when $p < 1$. In the sequel, we use the following notation for more clarity :

$$\|f_{\cdot,k}\| = \left(\sum_{t=1}^{T} \|f_{t,k}\|^2_{\mathcal{H}_k} \right)^{1/2}$$

3.1 The $\ell_1 - \ell_2$ case

The optimization problem related to the sparse multi-task SVM can be posed as follows :

$$\min_{f_1,\cdots,f_T} C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_{k=1}^{M} \|f_{\cdot,k}\|$$

where $H(f_t(x), y) = \max(0, 1 - y f_t(x))$ is the Hinge loss function and C the usual SVM parameter. This optimization problem is clearly convex but non-smooth because of the Hinge loss and the regularizer. However, the algorithmic difficulties are essentially due to the non-differentiability of $\|f_{\cdot,k}\|$ at 0. Similarly to recent works on MKL, we use a variational form of $\Omega_{1,2}$ which makes this latter differentiable at the expense of adding new variables to the optimization problem (Rakotomamonjy *et al.*, 2008) :

$$\begin{array}{ll} \min\limits_{f_1,\cdots,f_T,\mathbf{d}} & C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_k \frac{\|f_{\cdot,k}\|^2}{d_k} \\ \text{s.t} & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \end{array}$$

Here and in what follows, we take the convention that $\frac{u}{0} = 0$ if $u = 0$ and ∞ otherwise. After, expanding $\|f_{\cdot,k}\|^2$ and re-arranging the sums, we note that for a fixed $\mathbf{d}$ (vector with entries d_k), each task can be trained independently as made explicit through the following equivalent optimization problem :

$$\begin{array}{ll} \min\limits_{\mathbf{d}} & J(\mathbf{d}) = \sum_t J_t(\mathbf{d}) \\ \text{s.t} & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \end{array} \tag{3}$$

with

$$J_t(\mathbf{d}) = \min_{f_t} \; C \sum_i H(f_t(x_{i,t}), y_{i,t}) + \sum_k \frac{\|f_{t,k}\|^2}{d_k} \tag{4}$$

This latter formulation shows how our sparse multi-task SVM problem is strongly related to the MKL problem. At first, we remark that the Equations (3-4) boil down to be the MKL problem when only a single task is considered. When several tasks are in play, the vector $\mathbf{d}$ makes explicit that they are linked through their shared sparse kernel representation.

For solving this optimization problem, we build on the gradient-based MKL algorithm (Rakotomamonjy *et al.*, 2008). This MKL algorithm can be straightforwardly extended to our problem by noting that the minimization problem (4) yields the objective value of a SVM problem with kernel $K = \sum_k d_k K_k$. Indeed the minimization with respects to f_t in (4) is equivalent to the minimization over the functions $f_{t,k}, \forall k$

Algorithm 1 $\ell_1 - \ell_2$ sparse MTL solver

$d_k^1 = \frac{1}{M}$ for $k = 1, \cdots, M$.
for $n = 1, 2, \cdots$ **do**
  Solve each SVM task with $K = \sum_{k=1}^{M} d_k K_k$.
  Compute $\frac{\partial J}{\partial d_k}$ for $k = 1, \cdots, M$ as given in Equation (6).
  Compute descent direction D_n and optimal step γ_n such that $\mathbf{d}^{n+1} \leftarrow \mathbf{d}^n + \gamma_n D_n$.
  if stopping criterion **then**
    break
  end if
end for

and b_t according to the expression (1) of the t^{th} decision function. For instance, the optimality condition w.r.t. $f_{t,k}$ is :

$$f_{t,k}(\cdot) = d_k \sum_i \alpha_{i,t} y_{i,t} K_k(x_{i,t}, \cdot)$$

where the $\alpha_{i,t}$ are the Lagrange multipliers related to the classical SVM constraints embedded in the hinge loss. The same algebra for the bias b_t yields $\sum_i \alpha_{i,t} y_{i,t} = 0$. Therefore, it comes up the dual problem corresponding to (4) turns into

$$\begin{array}{ll} \min\limits_{\alpha_{i,t}} & \frac{1}{2} \sum_{i,j} \alpha_{i,t} \alpha_{j,t} y_{i,t} y_{j,t} \sum_k d_k K_k(x_{i,t}, x_{j,t}) - \sum_i \alpha_{i,t} \\ \text{s.t} & \alpha_{i,t} y_{i,t} = 0, \quad \text{and} \quad 0 \leq \alpha_{i,t} \leq C \quad \forall i \end{array} \tag{5}$$

Then since the objective function of the sparse multi-task learning given in Equation (3) is just a sum of single task SVM objective value, its gradient is simply :

$$\nabla_{d_k} J(\mathbf{d}) = -\frac{1}{2} \sum_t \sum_{i,j} \alpha_{i,t}^\star \alpha_{j,t}^\star y_{i,t} y_{j,t} K_k(x_{i,t}, x_{j,t}) \tag{6}$$

where the $\alpha^\star$ are the optimal alpha's that minimize Equation (5). Equations (5) and (6) provide the ingredients to apply the recipes of SimpleMKL algorithm to the sparse multi-task learning (we refer the reader to the aforementioned paper for more details about the machinery of MKL). The different steps of our $\ell_1 - \ell_2$ MTL solver are briefly summarized in Algorithm 1.

Regarding convergence and complexity of this algorithm, we can state that they are strongly related to the ones of gradient-descent based MKL. Hence, we can just remind that convergence towards the problem global minimum is ensured if each SVM task is exactly solved (*e.g* with 0 duality gap) which means that the gradient in Equation (6) is exact. The algorithm 1 complexity is then of the same order of SimpleMKL ones. Indeed, the main difference is that T SVM tasks have to be solved and that the gradient computation involves the T tasks.

3.2 The $\ell_p - \ell_2$ (with $p < 1$) case

Now that we are able to solve the sparse MTL problem using a $\ell_1 - \ell_2$ mixed norms, we propose an algorithm which solves the non-convex case where $\ell_p - \ell_2$ (with $p < 1$

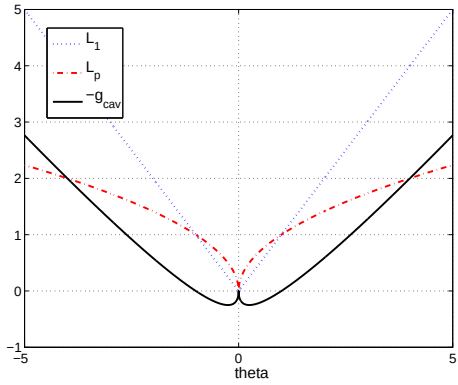

FIG. 1 – Difference of convex functions representation for an ℓ_p quasi-norm (here $p = 0.5$).

). For this novel situation, let rewrite the regularization term as :

$$\Omega_{p,2} = \sum_{k=1}^{M} g(\|f_{\cdot,k}\|) \tag{7}$$

where the upper level penalty function is $g(u) = u^p, u > 0$ with $p < 1$. Clearly, this function is non-convex. To address this issue, we investigate the use of DC programming (Horst & Thoai, 1999) which is a general framework for optimizing non-convex objective functions that can be expressed as a difference of convex functions (or a sum of a convex and concave functions i.e. $\min_\theta J(\theta) = \min_\theta J_{vex}(\theta) + J_{cav}(\theta)$). The trick of the DC algorithm is broadly used in machine learning (see for e.g. (Sriperumbudur *et al.*, 2007) for recent publication). For situations where the concave function is differentiable, the DC algorithm is an iterative procedure where at the i^{th} iteration, one optimizes the problem :

$$\theta^{(i+1)} = \min_\theta J_{vex}(\theta) + \langle \nabla_\theta J_{cav}(\theta^{(i)}), \theta - \theta^{(i)} \rangle$$

until convergence. For our multi-task problem, we propose a decomposition that enables us to use the $\ell_1 - \ell_2$ MTL solver. Indeed, we suggest the following decomposition :

$$g(u) = g_{vex}(u) + g_{cav}(u) = u - (u - u^p)$$

which leads to

$$\begin{array}{rcl} J_{vex} & = & C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_k \|f_{\cdot,k}\| \\ J_{cav} & = & \sum_k (-\|f_{\cdot,k}\| + \|f_{\cdot,k}\|^p) \end{array}$$

An illustration of such a decomposition in one dimension case is given in Figure 1 for $u = |\theta|$. Notice that here, we are interested only on the positive part on these curves.

Now according to this decomposition, we have :

$$\nabla_{f_{t,k}} J_{cav} = \left(-1 + p\|f_{\cdot,k}\|^{p-1}\right) \nabla_{f_{t,k}} \|f_{\cdot,k}\|$$

where the derivative $\nabla_{f_{t,k}} \|f_{\cdot,k}\| = \frac{f_{t,k}(\cdot)}{\|f_{\cdot,k}\|}$ is easily derived. Therefore, at each DC iteration, after tedious algebras and owing to the first-order approximation $\frac{\langle f_{t,k}^{(i)}, f_{t,k}\rangle}{\|f_{t,k}^{(i)}\|} = \|f_{t,k}\|$, we show that :

$$\min_{f_1,\cdots,f_T} C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_k p \frac{\|f_{\cdot,k}\|}{\|f_{\cdot,k}^{(i)}\|^{1-p}}$$

This latter equation demonstrates that, in order to solve the non-convex $\ell_p - \ell_2$ case using a DC programming approach, one needs to solve iteratively a weighted $\ell_1 - \ell_2$ multi-task problem

$$\min_{f_1,\cdots,f_T} C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_{k=1}^{M} \beta_k \|f_{\cdot,k}\| \tag{8}$$

where β_k are some fixed coefficients, which in our case would depend on the iteration and are defined at the i^{th} iteration as :

$$\beta_k = \frac{p}{\|f_{\cdot,k}^{(i)}\|^{1-p}}, \quad \forall\, k = 1, \cdots, M \tag{9}$$

This definition of the β_k requires implicitly the positivity of $\|f_{\cdot,k}\|$. To ensure this condition, a small term ϵ is added to $\|f_{\cdot,k}\|$ in (7). Hence, this involves to consider rather $\beta_k = \frac{p}{\epsilon + \|f_{\cdot,k}^{(i)}\|^{1-p}}$. This trick suggested as well by Candès *et al.* (2008) avoids numerical instabilities.

Now, the equivalent optimization problem with smooth regularization is simply :

$$\begin{array}{ll} \min\limits_{f_1,\cdots,f_T,\mathbf{d}} & C \sum_{t,i} H(f_t(x_{i,t}), y_{i,t}) + \sum_k \beta_k^2 \frac{\|f_{\cdot,k}\|^2}{d_k} \\ \text{s.t} & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \end{array} \tag{10}$$

Note that the optimality conditions of this problem with respects to $f_{t,k}$ is simply given by the expression $f_{t,k}(\cdot) = \frac{d_k}{\beta_k^2} \sum_i \alpha_{i,t} y_{i,t} K_k(x_{i,t}, \cdot)$.

Consequently, at each DC iteration, we have to solve a weighted sparse MTL problem, where the weights are applied to the basis kernels. Hence, Equation (10) can be solved using the $\ell_1 - \ell_2$ algorithm just by replacing the kernel $K_k(x, x')$ with $\frac{1}{\beta_k^2} K_k(x, x')$.

Details of the $\ell_p - \ell_2$ problem solver are given in Algorithm 2. About its complexity, we can state that since the $\ell_p - \ell_2$ algorithm is based on *iter* iterations of the $\ell_1 - \ell_2$ algorithm (after appropriate rescaling of the kernels), its complexity can be approximated as *iter* times the $\ell_1 - \ell_2$ algorithm complexity. However, in order to speed-up convergence for $\ell_p - \ell_2$, warm-starting the $\ell_1 - \ell_2$ with results from previous iteration can be

Algorithm 2 $\ell_p - \ell_2$ sparse MTL solver

$\beta_k = 1$ for $k = 1, \cdots, M$
Compute K_k kernel matrices for all tasks
repeat
  $K_k^\beta \leftarrow \frac{K_k}{\beta_k^2}$ for all k
  Solve $\ell_1 - \ell_2$ MTL problem with kernels K_k^β
  Update β_k using Equation (9)
until convergence of the β's

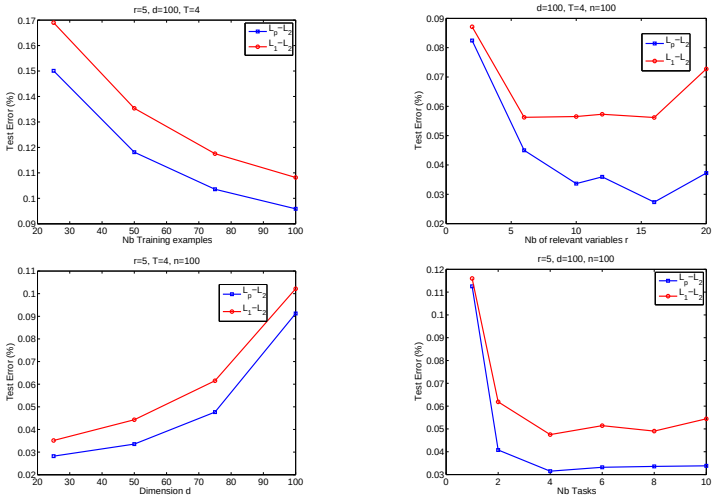

FIG. 2 – Performance comparisons between $\ell_1 - \ell_2$ and $\ell_p - \ell_2$ penalty for different experimental situations. For each experimental situations, we have kept fixed all except one of parameters : number of training examples n, number of relevant variables r, problem dimension d and number of tasks T. top-left) varying n. top-right) varying r. bottom-left) varying d bottom-right) varying T.

beneficial. Indeed, one may expect for instance, that many of the vanishing coefficients d_k at a given iteration will stay at zero at the next iteration.

The local convergence of Algorithm 2 is guaranteed. Indeed, the DC programming approach proceeds by surrogating the concave part of the objective function with its affine majorization at each iteration. Therefore, the minimized function decreases until a guaranteed convergence to at least a local minimum (Horst & Thoai, 1999).

TAB. 1 – Average AUC performances of 4 different algorithms on the BCI datasets. The number of variables that have been kept in the decision function is also given.

Algorithms	AUC	# variables
MTL_1	85.72 ± 1.8	192 ± 11
$\text{MTL}_{0.5}$	86.37 ± 1.3	43 ± 6
FullMKL	86.17 ± 1.8	214 ± 12
SepMKL	84.15 ± 1.8	272 ± 13

4 Numerical experiments

4.1 Results on BCI datasets

Here we illustrate the usefulness of sparse Multi-Task learning on a Brain-Computer Interface problem. Indeed, sparse MTL can be very relevant to BCI because of the need of channel/variable selection and because of the data non-stationarity with respects to different subjects or even with respects to different acquisition sessions for a single subject.

The dataset we use is the training set of P300 Speller dataset from BCI 2003 competition and we treat the problem as a single trial classification of EEG signals. Such a dataset is composed of 11 acquisition sessions for which a subject has been asked to spell words of 3 to 5 characters. For each session, 540 to 900 EEG signals (180 for a character) have been acquired and paired with a positive or negative stimuli responses. After preprocessing as in Rakotomamonjy *et al.* (2005), the signal becomes a vector of dimension 896 (14 time frames for each of the 64 channels).

Here, sparse MTL is particularly relevant because channel selection is known to enhance BCI classification performance and furthermore, we believe that MTL can help handling inter-session variabilities. For instance, for the same problem, Rakotomamonjy *et al.* (2005) use an ensemble of linear SVMs where each SVM has been trained independently and using only examples belonging to the same acquisition sessions. Here, we train these SVMs using our sparse MTL approach and thus we impose that all linear SVMs share the same sparsity profile.

The experimental protocol is then the following. We have considered only 4 acquisition sessions, thus 4 tasks. For these sessions, we have randomly picked 180 training examples and used the remaining as testing examples. C has been fixed to 10 which is small enough for achieving good sparsity. This overall procedure has been repeated 10 times.

Table 1 summarizes the average performance of 4 different algorithms : a MKL SVM trained on all training examples (FullMKL), an ensemble of MKL SVM (SepMKL) where each SVM has been trained according to data from a single session (this approach is equivalent to the state-of-the art method) , our sparse $\ell_p - \ell_2$ MTL with respectively $p = 1$ and $p = 0.5$. Algorithm performance has been evaluated according to AUC obtained by feeding the test set to all SVMs and by summing all obtained scores. The final score is then used for computing AUC.

Interpreting these results tells us that performance of the 4 algorithms are equivalent with a slight advantage for sparse MTL with $p = 0.5$. Interestingly, taking into account a relation between tasks allows slightly better performances than training tasks independently. The most interesting point for our sparse MTL approach is the performance we achieve using only about 5% of the variables. For real-time application of BCI, such a dimensionality reduction is of primary importance.

5 Conclusion

In this paper, we investigated the use of mixed-norms for Multi-Task SVM with joint sparsity constraint. After having proposed a class of penalty function based on a $\ell_p - \ell_2$ norm, we first derive an algorithm which addresses the convex optimization problem when $p = 1$. For the case $p < 1$, we fitted the optimization problem into the DC programming framework, and proposed an iterative reweighted version of the $\ell_1 - \ell_2$ algorithm. One interesting point of the algorithms we propose is that they can both take advantage of any progress made in SVM and MKL efficiency. Experimental results brought evidence that $\ell_p - \ell_2$ penalties lead to enhanced performance and better sparsity especially in situations where a large number of variables are in play.

Future works aim at proposing a generic algorithm that can handle the general situation of $\ell_p - \ell_q$ norm and at theoretically analyzing the consistency of our reweighted algorithm.

Références

ARGYRIOU A., EVGENIOU T. & PONTIL M. (2008). Convex multi-task feature learning. *Machine Learning, to appear*.

BACH F. (2008). Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, **9**, 1179–1225.

BI J., XIONG T., YI S., DUNDAR M. & RAO B. (2008). An improved multi-task learning approach with applications in medical diagnosis. In *Proceedings of the 18th European Conference on Machine Learning*.

CANDÈS E., WAKIN M. & BOYD S. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Analysis and Applications*, **14**, 877–905.

CARUANA R. (1997). Multi-task learning. *Machine Learning*, **28**, 41–75.

COTTER S., RAO B., ENGAN K. & KREUTZ-DELGADO K. (2005). Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, **53**(7), 2477–2488.

EVGENIOU T. & PONTIL M. (2004). Regularized multi-task learning. In *Proceedings of the tenth Conference on Knowledge Discovery andData minig*.

HORST R. & THOAI N. V. (1999). Dc programming : overview. *Journal of Optimization Theory and Applications*, **103**, 1–41.

JEBARA T. (2004). Multi-task feature and kernel selection for svms. In *Proceeding of the 21st International Conference on Machine Learning*.

KATO T., KASHIMA H., SUGIYAMA M. & ASAI K. (2008). Multi-task learning via conic programming. In *Advances in Neural Information Processing Systems*.

KNIGHT K. & FU W. (2000). Asymptotics for lasso-type estimators. *Annals of Statistics*, **28**, 1356–1378.

LIU H., LAFFERTY J. & WASSERMAN L. (2009). Non parametric regression and classification with joint sparsity constraints. In D. KOLLER, D. SCHUURMANS, Y. BENGIO & L. BOTTOU, Eds., *Advances in Neural Information Processing Systems 21*.

OBOZINSKI G., TASKAR B. & JORDAN M. (2007). *Multi-task feature selection*. Rapport interne, UC Berkeley Technical Report.

QUATTONI A., COLLINS M. & DARRELL T. (2008). Transfer learning for image classification with sparse prototype representations. In *Proceedings of CVPR*.

RAKOTOMAMONJY A., BACH F., GRANDVALET Y. & CANU S. (2008). SimpleMKL. *Journal of Machine Learning Research*, **9**, 2491–2521.

RAKOTOMAMONJY A., GUIGUE V., MALLET G. & ALVARADO V. (2005). Ensemble of SVMs for improving brain-computer interface p300 speller performances. In *15th International Conference on Artificial Neural Networks*.

SINDHWANI V., NIYOGI P. & BELKIN M. (2005). Beyond the point cloud : from transductive to semi-supervised learning. In *Proceedings of International Conference on Machine Learning*.

SRIPERUMBUDUR B. K., TORRES D. A. & LANCKRIET G. (2007). Sparse eigen methods by d.c. programming. In *Proceedings of the 24 th International Conference on Machine Learning*.

SZAFRANSKI M., GRANDVALET Y. & RAKOTOMAMONJY A. (2008). Composite kernel learning. In *Proceedings of the 22nd International Conference on Machine Learning*.

TIBSHIRANI R. (1995). Regression selection and shrinkage via the lasso. *Journal of the Royal Statistical Society*, p. 267–288.

TROPP J. (2006). Algorithms for simultaneous sparse approximation. part ii : Convex relaxation. *Journal of Signal Processing*, **86**, 589–602.

XIONG T., BI J., RAO B. & CHERKASSKY V. (2006). Probabilistic joint feature selection for multi-task learning. In *Proceedings of SIAM International Conference on Data Mining*.

YU K., TRESP V. & SCHWAIGHOFER A. (2005). Learning gaussian processes from multiple tasks. In *Proceeding of the 22nd International Conference on Machine Learning*.

ZHAO P., ROCHA G. & YU B. (to appear). The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*.

ZOU H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**(476), 1418–1429.