SVM MULTI-TASK LEARNING AND NON-CONVEX SPARSITY MEASURE

FLAMARY Remi[2], RAKOTOMAMONJY Alain[2], GASSO Gilles[1] and CANU Stéphane[2]

[1] NEC Labs America
4, Independence way, Princeton, NJ 08540
[2]LITIS, EA 4108
Avenue de l'Université
F-76801 St Etienne du Rouvray Cedex - France
Email: gilles@necs-lab.com, {remi.flamary, alain.rakoto, stephane.canu}@insa-rouen.fr

Multi-task Learning

- Assume T classification tasks with T datasets $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1,\cdots,n_t}$ where $t = 1, \cdots, n_t$, $x_i \in \mathcal{X}$, $y_i \in \{-1, 1\}$
- Tasks are considered similar enough or related in a certain sense
- Aim: learn the decision functions $f_t(x), t = 1, \cdots, T$ in a joint manner
- How to do it? Tasks share a common subset of relevant features
- Way to ensure this constraint? Use adequate regularization that favors joint features sparsity pattern across tasks
- Our contributions
  - Application of multi-task learning principle to SVM framework by the selection of joint relevant kernels (multiple kernel learning coupled with multi-task learning)
  - Extension to handle non-convex regularisation

SVM multi-task learning: convex problem formulation

- $\min_{f_1,\cdots,f_T} \quad C \cdot \sum_{t=1}^T \sum_{i \in \mathcal{D}_t} L(f_t(x_i^t), y_i^t) + \Omega(f_1, \cdots, f_T) \qquad (1)$

  $L(y, f(x)) = \max(0, 1 - yf(x))$: hinge loss function and $\Omega(f_1, \cdots, f_T)$: joint sparsity regularizer
- **Multiple kernel framework**

  Each decision function is expressed as $f_t(x) = \sum_{k=1}^M f_{t,k}(x) + b_t$

  $f_{\cdot,k}$: function belonging to the Hilbert space $\mathcal{H}_k$ induced by kernel K_k
- **Joint sparsity regularization**

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

  Equivalent to $\ell_1 - \ell_2$ penalization (group lasso type regularization)

  Convex regularization $\Longrightarrow$ optimization problem (1) convex

SVM multi-task, multiple kernel learning

- Let $\|f_{\cdot,k}\| = \left(\sum_{t=1}^T \|f_{t,k}\|^2_{\mathcal{H}_k}\right)^{1/2}$. The variational formulation is

$$\begin{aligned} \min_{f_1,\cdots,f_T,\mathbf{d}} \quad & C \sum_{t=1}^T \sum_{i \in \mathcal{D}} L(f_t(x_i^t), y_i^t) + \sum_{k=1}^M \frac{\|f_{\cdot,k}\|}{d_k} \\ \text{s.t} \quad & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \end{aligned}$$

  Variables d_k: extra-parameters introduced to cope with the block-norm regularization. The values of d_k stress the importance of the corresponding kernels K_k in the SVM solution. $d_k = 0$ means kernel K_k discarded from the solution.
- Optimization problem

$$\begin{aligned} \min_{\mathbf{d}} \quad & J(\mathbf{d}) = \sum_t J_t(\mathbf{d}) \\ \text{s.t} \quad & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \qquad (2) \end{aligned}$$

$$\text{with} \quad J_t(\mathbf{d}) = \min_{f_t} \; C \sum_{i \in \mathbf{D}_t} L(f_t(x_i^t), y_i^t) + \sum_k \frac{\|f_{t,k}\|^2}{d_k}$$

- The parameters d_k are optimized by a projected gradient algorithm
- Knowing $\mathbf{d}$, each decision function f_t is retrieved from the solution of the SVM

$$\begin{aligned} \max_{\alpha_i^t} \quad & -\tfrac{1}{2} \sum_{i,j} \alpha_i^t \alpha_j^t y_i^t y_j^t \sum_k d_k K_k(x_i^t, x_j^t) + \sum_i \alpha_i^t \\ \text{s.t} \quad & \sum_i \alpha_i^t y_i^t = 0, \quad \text{and} \quad 0 \leq \alpha_i^t \leq C \quad \forall i \end{aligned}$$

Algorithm: $\ell_1 - \ell_2$ sparse Multi-task learning solver

Set $\mathbf{d}^1 = \frac{1}{M}\mathbb{1}$
for $n = 1, 2, \cdots$ **do**
  Solve each SVM task with kernel $K = \sum_{k=1}^M d_k K_k$.
  Compute the gradient $\frac{\partial J}{\partial d_k}$ for $k = 1, \cdots, M$ as

$$\nabla_{d_k} J(\mathbf{d}) = -\frac{1}{2} \sum_{t=1}^T \sum_{i,j} \alpha_i^t \alpha_j^t y_i^t y_j^t K_k(x_i^t, x_j^t)$$

  Compute descent direction D_n and optimal step γ_n such that $\mathbf{d}^{n+1} \leftarrow \mathbf{d}^n + \gamma_n D_n$ and constraints (2) satisfied
  if stopping criterion **then**
    break
  end if
end for

Handling non-convex joint sparsity regularizer

Non-convex regularization

- Instead of the $\ell_1 - \ell_2$ penalty, consider a non-convex "pseudo-norm" $\ell_p - \ell_2$ penalty with $p < 1$
- Aim: emphasize the sparse behavior of the solution
- Proposed regularization closely related to the spirit of non-convex group lasso algorithms that was issued from consistency results of the convex group lasso
- Non-convex regularizer: $\Omega(f_1, \cdots, f_T) = \sum_{k=1}^M g(\|f_{\cdot,k}\|)$

  with $g(u) = u^p, \quad p < 1, \quad \text{and} \quad u \geq 0$

  Remark: any other penalty function $g(u)$ could be used as well

Proposed solution

- DC programming Principles
  - Assume the optimization problem $\min_\theta J(\theta) = \min_\theta J_1(\theta) - J_2(\theta)$
  - Solve iteratively $\theta^{(i+1)} = \text{argmin}_\theta \, J_1(\theta) - \langle \nabla_\theta J_2(\theta^{(i)}), \theta - \theta^{(i)} \rangle$ until convergence
- Application: use of the decompostion $g(u) = u - (u - u^p)$
- It leads to $J_1 = C \sum_{t,i} L(f_t(x_i^t), y_i^t) + \sum_k \|f_{\cdot,k}\|$ and $J_2 = \sum_k(-\|f_{\cdot,k}\| + \|f_{\cdot,k}\|^p)$
- Applying the DC algorithm, the non-convex joint sparsity optimization problem boils down to **solve iteratively a reweigthed $\ell_1 - \ell_2$ multi-task problem**

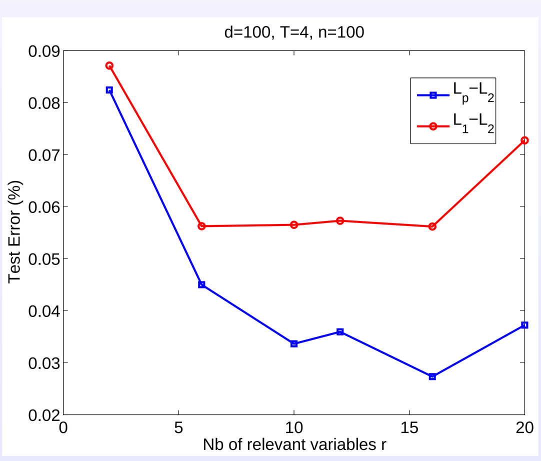
$$\begin{aligned} \min_{f_1,\cdots,f_T,\mathbf{d}} \quad & C \sum_t \sum_{i \in \mathcal{D}} L(f_t(x_i^t), y_i^t) + \sum_k \beta_k \frac{\|f_{\cdot,k}\|}{d_k} \\ \textbf{s.t} \quad & \sum_k d_k = 1, \quad d_k \geq 0 \quad \forall k \end{aligned}$$

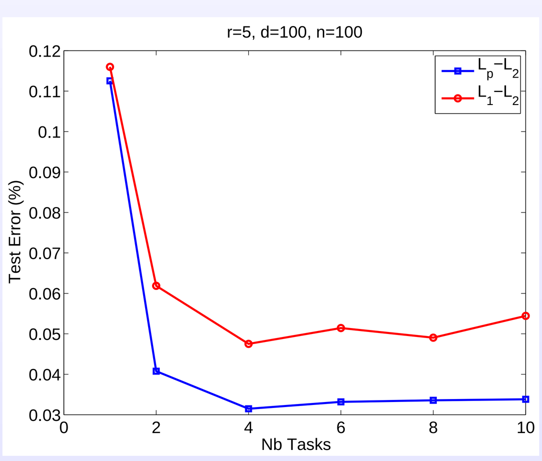
- At each iteratation, the weights are given by $\beta_k = \frac{p}{\|f^{(i)}_{\cdot,k}\|^{1-p}}$

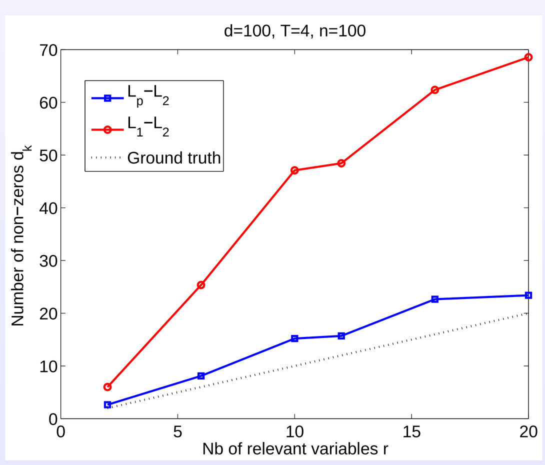
Experimental results

Example 1: Toy problem

- T binary classification tasks with n samples $x \in \mathbb{R}^d$ for each task
- The classes follow gaussian distributions with means μ, $-\mu$ and random covariance matrix in $\mathbb{R}^r$ where r is the number of relevant variables. The remaining $d - r$ variables are generated randomly and are considered as spurious variables
- Kernels: each dimension defined a kernel K_k, $\quad k = 1, \cdots, d$







Example 2: BCI dataset

- P300 Speller dataset acquired from 11 sessions
- Each session characterized by 400 to 950 EEG signals issued from 64 channels
- After preprocessing, 896 variables are generated
- Tasks: 4 acquisitions sessions (with the goal to handle inter-session variability)

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

Higher AUC is, better is the algorithm

MTL_1 and $\text{MTL}_{0.5}$: multi-task learning with $p = 1$ and $p = 0.5$

FullMKL: multiple kernel SVM trained on the entire available training set

SepMKL: tasks are trained separately