Unveiling Groups of Related Tasks in Multi-Task Learning

Jordan Frecon[1], Saverio Salzo[1], Massimiliano Pontil[1,2]

[1] CSML - Istituto Italiano di Tecnologia
[2] Dept of Computer Science - University College London

25th International Conference on Pattern Recognition, Milan, Italy

Multi-task learning

Setting: T linear regression tasks

find w_1	find w_2	find w_3	find w_4	find w_5	find w_6	$\cdots$	find w_T
$y_1 \approx X_1 w_1$	$y_2 \approx X_2 w_2$	$y_3 \approx X_3 w_3$	$y_4 \approx X_4 w_4$	$y_5 \approx X_5 w_5$	$y_6 \approx X_6 w_6$	$\cdots$	$y_T \approx X_T w_T$

$$W = [w_1 \cdots w_T]$$

low-rank

$$\hat{W} \in \underset{W=[w_1 \cdots w_T]}{\operatorname{argmin}} \sum_{t=1}^{T} \frac{1}{2} \|y_t - X_t w_t\|^2 + \lambda \|W\|_{\mathrm{tr}}$$

Multi-task learning

Setting: T linear regression tasks arranged in L groups of related tasks $\{\mathcal{G}_1, \dots, \mathcal{G}_L\}$

find w_1	find w_2	find w_3	find w_4	find w_5	find w_6	$\cdots$	find w_T
$y_1 \approx X_1 w_1$	$y_2 \approx X_2 w_2$	$y_3 \approx X_3 w_3$	$y_4 \approx X_4 w_4$	$y_5 \approx X_5 w_5$	$y_6 \approx X_6 w_6$	$\cdots$	$y_T \approx X_T w_T$

$W_{\mathcal{G}_1} = [w_1 w_2]$ low-rank

$W_{\mathcal{G}_2} = [w_3 w_4 w_5]$ low-rank

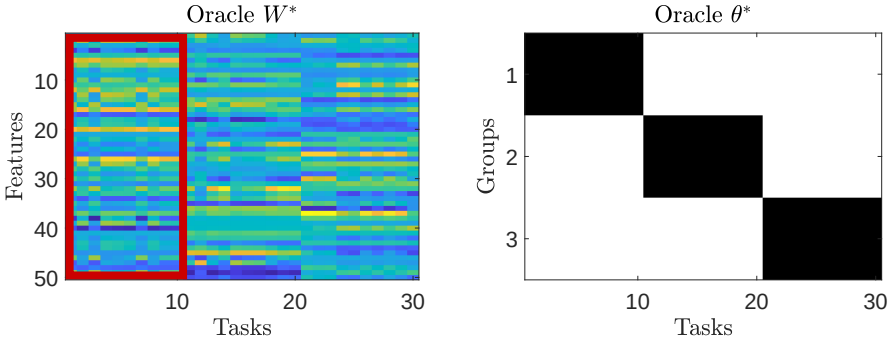
$W_{\mathcal{G}_3} = [w_6 \cdots w_T]$ low-rank

$$\hat{W} \in \underset{W=[w_1 \cdots w_T]}{\operatorname{argmin}} \sum_{t=1}^{T} \frac{1}{2} \|y_t - X_t w_t\|^2 + \lambda \sum_{l=1}^{L} \|W_{\mathcal{G}_l}\|_{\mathrm{tr}}$$

Issue: In practice we don't know how tasks are related $\rightarrow$ need to estimate $\{\mathcal{G}_1, \dots, \mathcal{G}_L\}$

Parametrization of related tasks

$$\hat{W}(\theta_1 \cdots \theta_L) \in \underset{W=[w_1 \cdots w_T]}{\operatorname{argmin}} \sum_{t=1}^{T} \frac{1}{2} \|y_t - X_t w_t\|^2 + \lambda \sum_{l=1}^{L} \|\theta_l \odot W\|_{\mathrm{tr}},$$



Goal: Estimation of the optimal group-structure θ^*

A Bilevel Programming Approach

Upper-level Problem:

$$\underset{[\theta_1 \cdots \theta_L] \in \Theta}{\text{minimize}} \, \mathcal{U}(\theta) := \sum_{t=1}^{T} \mathcal{E}_t(\hat{w}_t(\theta)) \qquad \textit{(e.g., validation error)}$$

where $\hat{W}(\theta) = \left[\hat{w}_1(\theta) \cdots \hat{w}_T(\theta)\right]$ solves

Lower-level Problem:

$$\underset{W=[w_1 \cdots w_T]}{\text{minimize}} \, \mathcal{L}(W, \theta) := \sum_{t=1}^{T} \frac{1}{2} \|y_t - X_t w_t\|^2 + \lambda \sum_{l=1}^{L} \|\theta_l \odot W\|_{\mathrm{tr}}$$

Difficulties:

- $\hat{W}(\theta)$ not available in closed form
- $\theta \mapsto \hat{W}(\theta)$ is nonsmooth [$\Rightarrow \mathcal{U}$ is nonsmooth]

Approximate Bilevel Problem

Upper-level Problem:

$$\underset{[\theta_1 \cdots \theta_L] \in \Theta}{\text{minimize}} \; \mathcal{U}_K(\theta) := \sum_{t=1}^{T} \mathcal{E}_t(w_t^{(K)}(\theta))$$

where $w_t^{(K)}(\theta) \to \hat{w}_t(\theta)$

Dual Algorithm:

$U^{(0)}(\theta)$ chosen arbitrarily
for $k = 0, 1, \ldots, K-1$
$\quad U^{(k+1)}(\theta) = \mathcal{A}(U^{(k)}(\theta), \theta)$ *dual update*
$W^{(K)}(\theta) = \mathcal{B}(U^{(K)}(\theta), \theta)$ *primal dual relationship*

Goals:

- Find $\mathcal{A}$ and $\mathcal{B}$ smooth [$\Rightarrow$ $\mathrm{w}^{(K)}$ is smooth $\Rightarrow$ $\mathcal{U}_K$ is smooth]
- Prove that the approximate bilevel scheme converges

.

Contributions

- **Bilevel framework for finding groups of related tasks**
- **Design of a dual forward-backward algorithm with Bregman distances such that**
  1. $\mathcal{A}$ and $\mathcal{B}$ are smooth $\Rightarrow$ $\mathcal{U}_K$ is smooth
  2. $\begin{cases} \min \mathcal{U}_K \to \min \mathcal{U} \\ \operatorname{argmin} \mathcal{U}_K \to \operatorname{argmin} \mathcal{U} \end{cases}$

Implementation of a projected gradient descent algorithm (and some variants)

$$\theta^{(q+1)} = \mathcal{P}_\Theta\big(\theta^{(q)} - \gamma \nabla \mathcal{U}_K(\theta^{(q)})\big)$$

Technicality: $\mathcal{U}_K$ involves generalized matrix functions $\to$ computing $\nabla \mathcal{U}_K$ requires attention

Numerical Experiment

Setting: $T = 30$ tasks arranged of 3 groups made of 10 tasks each.
$N = 10$ noisy observations and $P = 20$ features per task.
$\rightarrow$ Estimate and group the features into, at most, $L = 6$ groups.

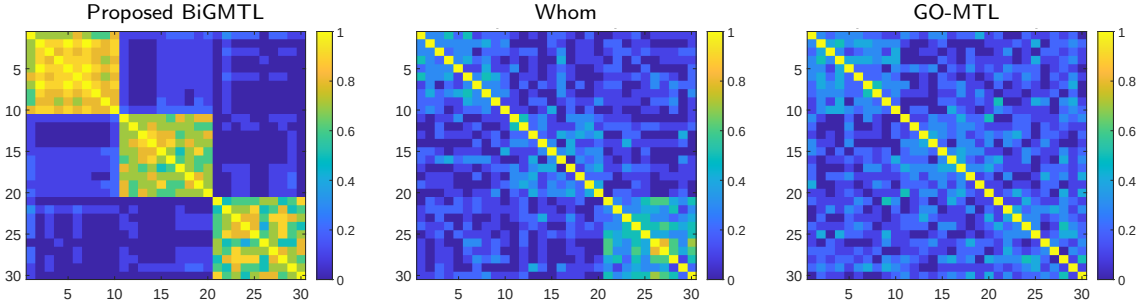


Figure 1: Mean group covariance matrix $\theta^\top \theta$ on the synthetic experiment. Only the proposed method manages to clearly estimate the three groups of tasks.

Conclusion

Thank You

A Matlab toolbox will be available at https://github.com/jordanFrecon