Bilevel Optimization of Hyperparameters: Application to Structure Discovery

Jordan Frecon

Apprentissage, LITIS, INSA Rouen

slides & more available at jordan-frecon.com

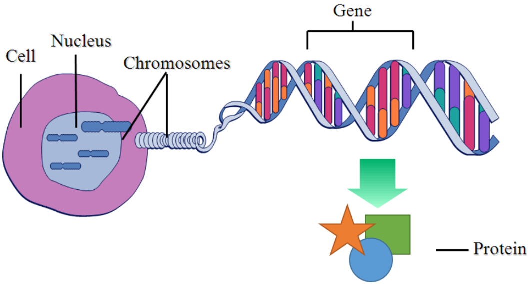
Example 1: Structured linear regression

Setting presented in
[Frecon et al. "Bilevel learning of the group Lasso structure". NeurIPS (2018)]

Motivation: genes expression analysis

Goal : Predict the function of proteins from regulatory patterns

Collaboration with Giorgio Valentini (Universita degli Studi di Milano)



Gene = long sequence with regulatory patterns (dictate gene expression)

Proteins perform various functions (transport, redox, binding ...)

Regulatory patterns

	R_1	R_2	R_3	...	R_P
	CTGAC	GGATC	GCAAG	...	ATCAG
Gene 1	1	1	0	...	0

neuron =

Protein sub-functions (Gene Ontology)

Go_1	Go_2	Go_3	...	Go_T
transport	redox	binding		
1	0	0	...	1

Sequences R_1 and R_2 are present in Gene 1

Gene 1 produces neurons

Neurons perform transport of electrons

Motivation: genes expression analysis

Regulatory patterns

	R_1	R_2	R_3	...	R_P
	CTGAC	GGATC	GCAAG	...	ATCAG
Gene 1	1	1	0	...	0
Gene 2	1	1	1	...	0

Protein sub-functions (Gene Ontology)

	Go_1	Go_2	Go_3	...	Go_T
	transport	redox	binding		
neuron	1	0	0	...	1
hormone	1	1	0	...	0

Sequences R_1, R_2 and R_3 are present in Gene 2

Gene 2 produces hormones

Hormones perform transport of particles and reduction–oxidation

Regulatory patterns

	R_1 CTGAC	R_2 GGATC	R_3 GCAAG	...	R_P ATCAG
Gene 1	1	1	0	...	0
Gene 2	1	1	1	...	0
Gene 3	0	0	0	...	1

Protein sub-functions (Gene Ontology)

	Go_1 transport	Go_2 redox	Go_3 binding	...	Go_T
neuron	1	0	0	...	1
hormone	1	1	0	...	0
antibody	0	0	1	...	0

Sequence R_P is present in Gene 3

Gene 3 produces antibodies

Antibodies perform binding of particles

Motivation: genes expression analysis

Regulatory patterns

	R_1	R_2	R_3	...	R_P
	CTGAC	GGATC	GCAAG	...	ATCAG
Gene 1	1	1	0	...	0
Gene 2	1	1	1	...	0
Gene 3	0	0	0	...	1
	⋮	⋮	⋮	⋮	⋮
Gene N	1	1	0	...	1

$X \in \{0,1\}^{N \times P}$

neuron

hormone

antibody

Protein sub-functions (Gene Ontology)

Go_1	Go_2	Go_3	...	Go_T
transport	redox	binding		
1	0	0	...	1
1	1	0	...	0
0	0	1	...	0
⋮	⋮	⋮	⋮	⋮
0	0	1	...	1

$y = [y_1 \cdots y_T] \in \{0,1\}^{N \times T}$

Motivation: genes expression analysis

Regulatory patterns

	R_1 CTGAC	R_2 GGATC	R_3 GCAAG	... ...	R_P ATCAG
Gene 1	1	1	0	...	0
Gene 2	1	1	1	...	0
Gene 3	0	0	0	...	1
	⋮	⋮	⋮	⋮	⋮
Gene N	1	1	0	...	1

$X \in \{0,1\}^{N \times P}$

neuron

hormone

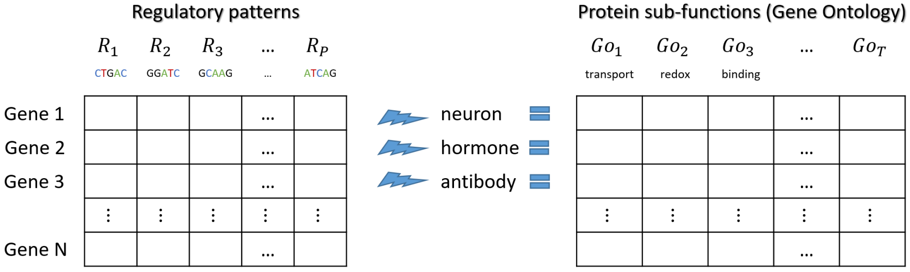
antibody

Protein sub-functions (Gene Ontology)

Go_1 transport	Go_2 redox	Go_3 binding	...	Go_T
1	0	0	...	1
1	1	0	...	0
0	0	1	...	0
⋮	⋮	⋮	⋮	⋮
0	0	1	...	1

$y = [y_1 \cdots y_T] \in \{0,1\}^{N \times T}$

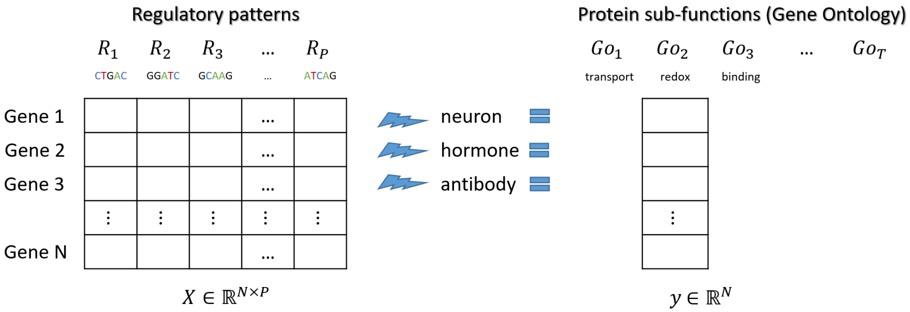
Goal 1: Predict each y_t from X



$X \in \mathbb{R}^{N \times P}$ $\quad$ $y = [y_1 \cdots y_T] \in \mathbb{R}^{N \times T}$

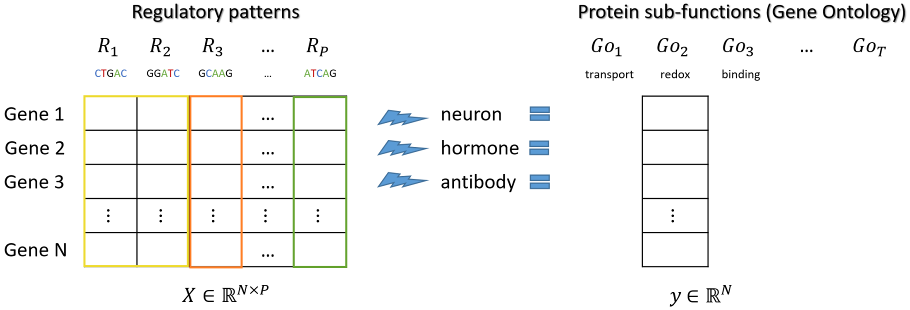
Goal 1: Predict each y_t from X

$\rightarrow$ to generalize: X and y are not only made of 0's and 1's



Goal 1: Predict y from X

$\rightarrow$ to generalize: X and y are not only made of 0's and 1's
$\rightarrow$ to simplify: we first assume that $T = 1$ and omit the index t



Goal 1: Predict y from X

→ to generalize: X and y are not only made of 0's and 1's
→ to simplify: we first assume that $T = 1$ and omit the index t

Goal 2: Discover if there exist some groups in X
ex: R_1 and R_2 are both equally relevant to predict y

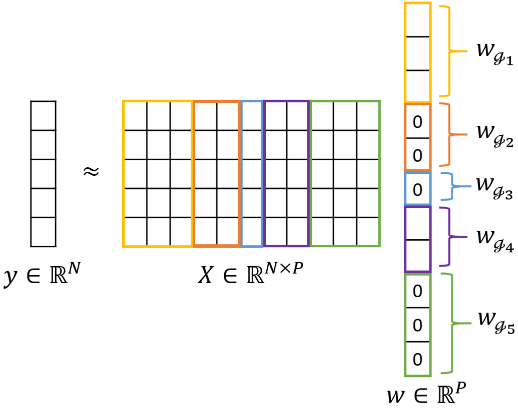
Model the observations

$\Longrightarrow$ linear model + Gaussian distribution: there exists w such that $y \sim \mathcal{N}(Xw, \sigma)$

Model the group structure

few groups of features in X are relevant to predict y

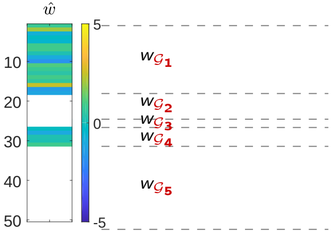
$\Longrightarrow$ group sparsity: some groups of variables in w are zero while others are non-zero



Optimization problem

Goal 1: Predict y from X Group Lasso [Yuan and Lin (2006)]

$$\hat{w}(\mathcal{G}_1, \dots, \mathcal{G}_L) = \underset{w \in \mathbb{R}^P}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|y - Xw\|^2}_{\propto -\log p(y|Xw)} + \underbrace{\lambda \sum_{l=1}^{L} \|w_{\mathcal{G}_l}\|_2}_{\text{enforces structure}}$$



L partitions $\mathcal{G}_1, \dots, \mathcal{G}_L$ of P features

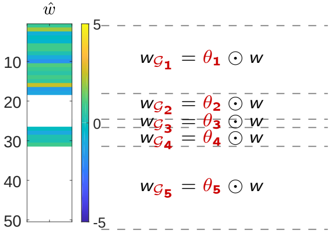
$$\mathcal{G}_l \subseteq \{1, \dots, P\}$$

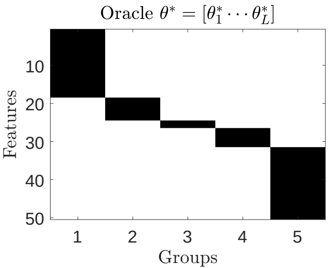
$$\mathcal{G}_l \cap \mathcal{G}_{l'} = \emptyset \text{ if } l \neq l'$$

$$\cup_{l=1}^{L} \mathcal{G}_l = \{1, \dots, P\}$$

Goal 1: Predict y from X Group Lasso [Yuan and Lin (2006)]

$$\hat{w}(\mathcal{G}_1,\ldots,\mathcal{G}_L) = \underset{w\in\mathbb{R}^P}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|y - Xw\|^2}_{\propto -\log p(y|Xw)} + \underbrace{\lambda \sum_{l=1}^{L} \|w_{\mathcal{G}_l}\|_2}_{\text{enforces structure}}$$



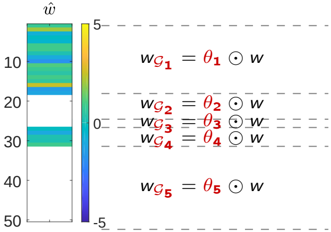


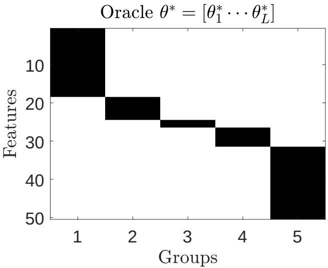
Mask $\theta_l = \{0,1\}^P$ of the l-th group

element-wise multiplication $\theta_l \odot w = [\theta_{l,1} w_1, \theta_{l,2} w_2, \ldots, \theta_{l,P} w_P]^\top$

Goal 1: Predict y from X Group Lasso [Yuan and Lin (2006)]

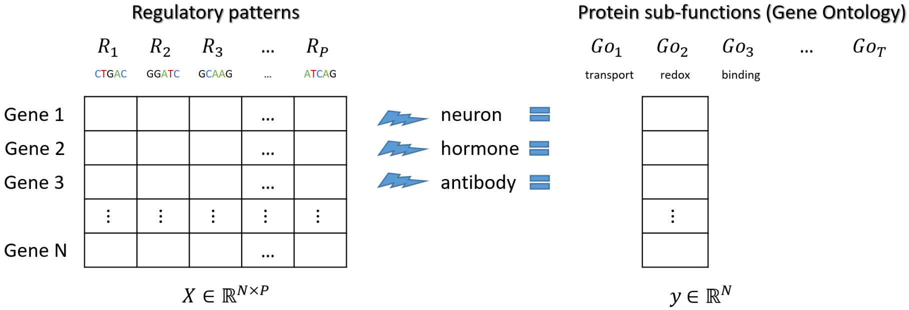
$$\hat{w}(\mathcal{G}_1, \ldots, \mathcal{G}_L) = \underset{w \in \mathbb{R}^P}{\text{argmin}} \; \underbrace{\frac{1}{2}\|y - Xw\|^2}_{\propto -\log p(y|Xw)} + \underbrace{\lambda \sum_{l=1}^{L} \|w_{\mathcal{G}_l}\|_2}_{\text{enforces structure}}$$





Goal 2: Discover the structure of X

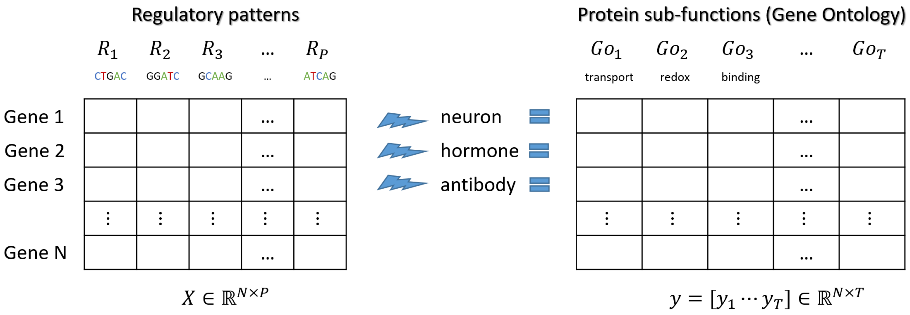
$\Longrightarrow$ finding $\{\mathcal{G}_1, \ldots, \mathcal{G}_L\}$ $\Longleftrightarrow$ learning the hyperparameter $\theta = [\theta_1 \cdots \theta_L] \in \{0, 1\}^{P \times L}$



Goal 1: Predict y from X

$\rightarrow$ to generalize: X and y are not only made of 0's and 1's
$\rightarrow$ to simplify: we first assume that $T = 1$ and omit the index t

Reminder



Goal 1: Predict y from X

$\rightarrow$ to generalize: X and y are not only made of 0's and 1's

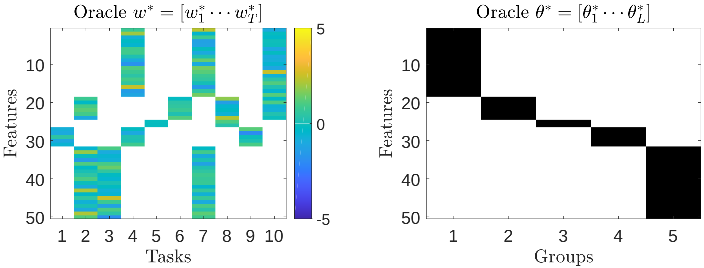
~~$\rightarrow$ to simplify: we first assume that $T = 1$ and omit the index t~~

Now we consider all T tasks

$$y \in \mathbb{R}^N \longrightarrow y \in \mathbb{R}^{N \times T}$$
$$w \in \mathbb{R}^P \longrightarrow w \in \mathbb{R}^{P \times T}$$

Multi-task setting : T tasks sharing the same group structure

$$(\forall t \in \{1, \ldots, T\}) \quad \hat{w}_t(\theta) \in \underset{w_t \in \mathbb{R}^P}{\operatorname{argmin}} \frac{1}{2}\|y_t - Xw_t\|^2 + \lambda \sum_{l=1}^{L} \|\theta_l \odot w_t\|_2,$$



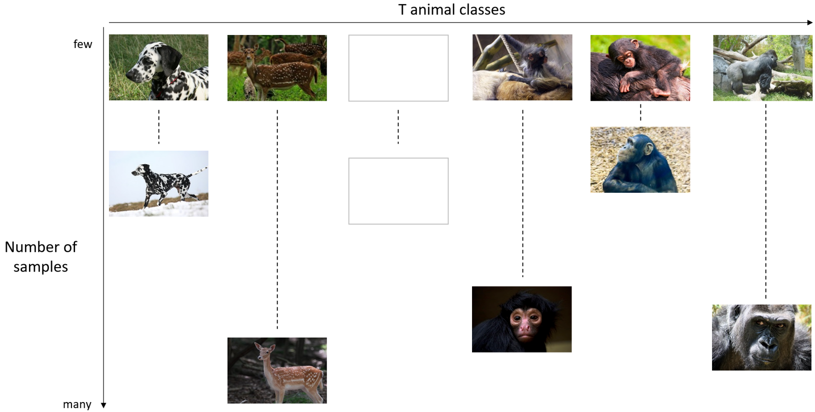
Some groups are relevant (non-zero) for some tasks and irrelevant (zero) for others

Example 2: Multi-task classification

Setting presented in

[Frecon et al. "Unveiling groups of related tasks in multi-task learning". ICPR (2020)]

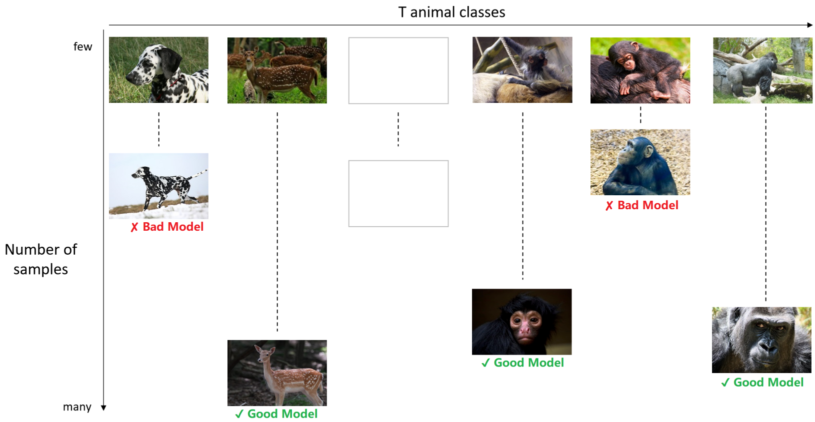
Motivation: animal recognition



$X \in \mathbb{R}^{N \times P}$ made of N vectorized image of P pixels
$y \in \mathbb{R}^{N \times T}$ such that $y_{i,t} = 1$ if X_i belongs to the t-th animal class, 0 otherwise.

Issue: some classes of animals with very few samples (e.g., dalmatians & chimpanzees)

Motivation: animal recognition

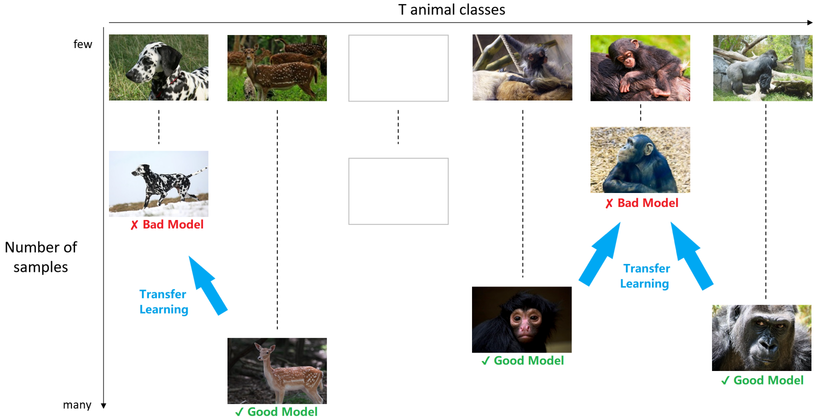


Naive idea: T binary classification tasks of one type of animal vs. all

logistic model: $(\forall t \in \{1, \dots, T\}), \exists w_t \in \mathbb{R}^P \mid y_t \sim \mathrm{Bernouilli}(p_t)$ with $p_t = \frac{1}{1+\exp(-Xw_t)}$

estimate $w_1, \dots, w_T \Rightarrow T$ independent binary logistic regressions

Motivation: animal recognition



Proposed idea: learn classifiers of similar animals jointly

logistic model: $(\forall t \in \{1, \ldots, T\}),\ \exists w_t \in \mathbb{R}^P \mid y_t \sim \mathrm{Bernouilli}(p_t)$ with $p_t = \frac{1}{1+\exp(-Xw_t)}$

estimate $w = [w_1 \cdots w_T] \Rightarrow$ Multi-task binary logistic regression

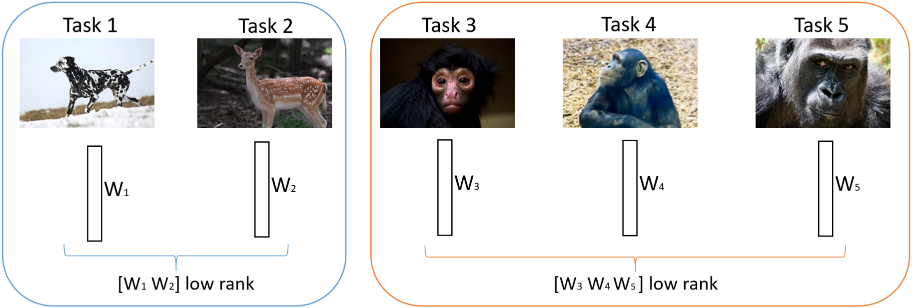
Model the observations

$$(\forall t \in \{1, \ldots, T\}),\ \exists w_t \in \mathbb{R}^P \mid y_t \sim \text{Bernouilli}(p_t) \text{ with } p_t = 1/(1 + \exp(-Xw_t))$$

Model task-relatedness

dalmatian (white + black spots) ≈ deer (brown + white spots)

$\Longrightarrow$ find $w_{\text{dalmatian}} \propto w_{\text{deer}}$ or more generally $[w_{\text{dalmatian}}\, w_{\text{deer}}]$ low-rank



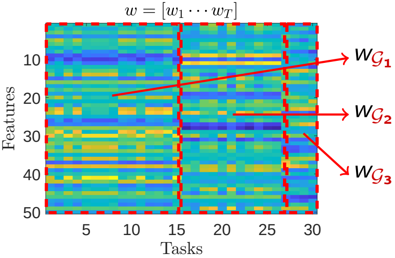
Group 1: "4 legs + spots"

Group 2: primates

Multi-task logistic regression: [Pong et al. (2010)] $\rightarrow$ here extended to L groups

$$\hat{w}(\mathcal{G}_1, \dots, \mathcal{G}_L) = \underset{w=[w_1 \cdots w_T]}{\operatorname{argmin}} \underbrace{\sum_{t=1}^{T} \log(1 - y_t \cdot X w_t)}_{\propto -\log p(y_t | X w_t)} + \lambda \underbrace{\sum_{l=1}^{L} \| w_{\mathcal{G}_l} \|_{\mathrm{tr}}}_{\text{enforces structure}}$$

Trace norm $\| \cdot \|_{\mathrm{tr}}$ = sum of singular values $\Rightarrow$ enforces low-rank



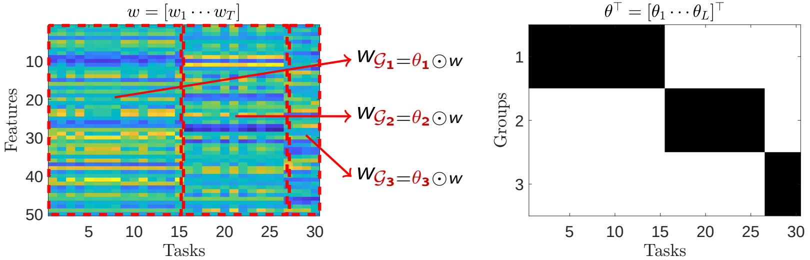
L partitions $\mathcal{G}_1, \dots, \mathcal{G}_L$ of T tasks

$$\mathcal{G}_l \subseteq \{1, \dots, T\}$$
$$\mathcal{G}_l \cap \mathcal{G}_{l'} = \emptyset \text{ if } l \neq l'$$
$$\cup_{l=1}^{L} \mathcal{G}_l = \{1, \dots, T\}$$

Multi-task logistic regression: [Pong et al. (2010)] $\rightarrow$ here extended to L groups

$$\hat{w}(\mathcal{G}_1, \ldots, \mathcal{G}_L) = \underset{w=[w_1 \cdots w_T]}{\operatorname{argmin}} \underbrace{\sum_{t=1}^{T} \log(1 - y_t \cdot X w_t)}_{\propto -\log p(y_t|Xw_t)} + \underbrace{\lambda \sum_{l=1}^{L} \|w_{\mathcal{G}_l}\|_{\mathrm{tr}}}_{\text{enforces structure}}$$

Trace norm $\|\cdot\|_{\mathrm{tr}}$ = sum of singular values $\Rightarrow$ enforces low-rank



Mask $\theta_l = \{0,1\}^T$ of the l-th group

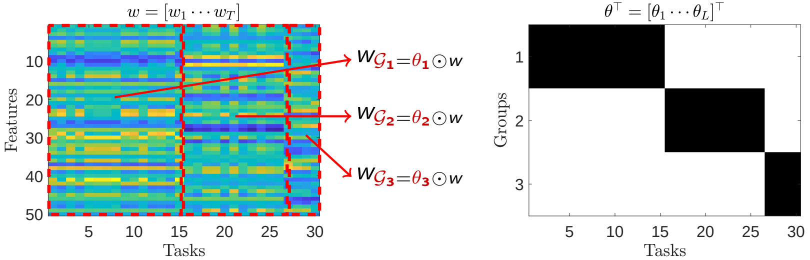
tasks-wise multiplication $\theta_l \odot w = [\theta_{l,1} w_1, \theta_{l,2} w_2, \ldots, \theta_{l,T} w_T]^\top$

Optimization problem

Multi-task logistic regression: [Pong et al. (2010)] $\rightarrow$ here extended to L groups

$$\hat{w}(\mathcal{G}_1,\dots,\mathcal{G}_L) = \underset{w=[w_1\cdots w_T]}{\operatorname{argmin}} \underbrace{\sum_{t=1}^{T} \log(1 - y_t \cdot X w_t)}_{\propto -\log p(y_t|Xw_t)} + \underbrace{\lambda \sum_{l=1}^{L} \|w_{\mathcal{G}_l}\|_{\mathrm{tr}}}_{\text{enforces structure}}$$

Trace norm $\|\cdot\|_{\mathrm{tr}}$ = sum of singular values $\Rightarrow$ enforces low-rank



When the optimal groups are unknown

$\Longrightarrow$ find $\{\mathcal{G}_1,\dots,\mathcal{G}_L\}$ $\Longleftrightarrow$ learn the hyperparameter $\theta = [\theta_1 \cdots \theta_L] \in \{0,1\}^{T\times L}$

Disclaimer

These are **motivating** examples to introduce the optimization problems
In practice, it is more complex ...

See the work of practitioners:

- Gene expressions [Higuera et al., (2015)]
- Animals images [Lambert et al. (2009)]
- Brain signals [Sabbagh et al. (2019)]

Proposed Framework

Groupwise regularized optimization problem

In both examples, the prediction phase requires to solve

$$\hat{w}(\theta_1, \ldots, \theta_L) = \underset{w}{\operatorname{argmin}} \left\{ \mathcal{L}(w;\theta) \triangleq \underbrace{\ell(y, \langle X, w\rangle)}_{\substack{\text{enforces model} \\ \propto -\log p(y|Xw)}} + \underbrace{\sum_{l=1}^{L} \rho_l(\theta_l \odot w)}_{\text{enforces structure}} \right\}$$

The structure is :

- encapsulated into $\theta = [\theta_1 \cdots \theta_L]$
- applied by the bilinear mapping $\odot$
- enforced by the **norms** ρ_l

Given the parameter matrix $w \in \mathbb{R}^{P\times T}$ made of P features and T tasks

- Grouping features (1st example) $\theta_l \in \{0,1\}^P$ and $\sum_{l=1}^{L} \theta_l = \mathbb{1}_P$
- Grouping tasks (2nd example) $\theta_l \in \{0,1\}^T$ and $\sum_{l=1}^{L} \theta_l = \mathbb{1}_T$

Learning the group structure θ

In many scenarios, the group structure θ is unknown or partly known

$$\Downarrow$$

Learning θ to improve results

Issue: difficult combinatorial problem
the number of possible partitions grows exponentially with the dimension
$\Rightarrow$ trying them all is out of reach

Idea: relax and optimize

indicators/masks $\theta = [\theta_1 \dots \theta_L]$ $\xrightarrow{\text{relaxation}}$ **probabilities** $\theta = [\theta_1 \dots \theta_L] \in \Theta$ simplex

θ_l mask of group l — θ_l probability to belong to group l

$\theta_{l,i} \in \{0, 1\}$ — $\theta_{l,i} \in [0, 1]$

Learning the group structure θ

In many scenarios, the group structure θ is unknown or partly known

$$\Downarrow$$

Learning θ to improve results

Issue: difficult combinatorial problem

the number of possible partitions grows exponentially with the dimension

$\Rightarrow$ trying them all is out of reach

Idea: relax and optimize

indicators/masks $\theta = [\theta_1 \dots \theta_L]$ $\xrightarrow{\text{relaxation}}$ **probabilities** $\theta = [\theta_1 \dots \theta_L] \in \Theta$ simplex

θ_l mask of group l — θ_l probability to belong to group l

$\theta_{l,i} \in \{0, 1\}$ — $\theta_{l,i} \in [0, 1]$

Learning the group structure θ

In many scenarios, the group structure θ is unknown or partly known

$$\Downarrow$$

Learning θ to improve results

Issue: difficult combinatorial problem
the number of possible partitions grows exponentially with the dimension
$\Rightarrow$ trying them all is out of reach

Idea: relax and optimize

indicators/masks		probabilities
indicators/masks $\theta = [\theta_1 \dots \theta_L]$	$\xrightarrow{\text{relaxation}}$	**probabilities** $\theta = [\theta_1 \dots \theta_L] \in \Theta$ simplex
θ_l mask of group l		θ_l probability to belong to group l
$\theta_{l,i} \in \{0, 1\}$		$\theta_{l,i} \in [0, 1]$

Optimizing the probabilities θ

We would like to find the groups $\theta \in \Theta$ such that the structured predictor

$$\hat{w}(\theta) = \operatorname*{argmin}_{w} \left\{ \mathcal{L}(w;\theta) \triangleq \ell(y, \langle X, w\rangle) + \sum_{l=1}^{L} \rho_l(\theta_l \odot w) \right\}$$

generalizes well to unseen data

Idea: Find θ such that $\hat{w}(\theta)$ minimizes the validation error $\mathcal{E}(\hat{w}(\theta)) = \ell(y^{\mathrm{val}}, \langle X^{\mathrm{val}}, \hat{w}(\theta)\rangle)$
$\Rightarrow$ type of continuous cross-validation

Bilevel Problem

$$\min_{\theta \in \Theta} \mathcal{E}(\hat{w}(\theta)) \qquad \text{s.t.} \qquad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

Optimizing the probabilities θ

We would like to find the groups $\theta \in \Theta$ such that the structured predictor

$$\hat{w}(\theta) = \operatorname*{argmin}_{w} \Big\{ \mathcal{L}(w;\theta) \triangleq \underbrace{\ell(y, \langle X, w \rangle)}_{\text{training error}} + \sum_{l=1}^{L} \rho_l(\theta_l \odot w) \Big\}$$

generalizes well to unseen data

Idea: Find θ such that $\hat{w}(\theta)$ minimizes the validation error $\mathcal{E}(\hat{w}(\theta)) = \ell(y^{\mathrm{val}}, \langle X^{\mathrm{val}}, \hat{w}(\theta) \rangle)$
$\Rightarrow$ type of continuous cross-validation

Bilevel Problem

$$\min_{\theta \in \Theta} \mathcal{E}(\hat{w}(\theta)) \quad \text{s.t.} \quad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

Optimizing the probabilities θ

We would like to find the groups $\theta \in \Theta$ such that the structured predictor

$$\hat{w}(\theta) = \operatorname*{argmin}_{w} \Big\{ \mathcal{L}(w;\theta) \triangleq \underbrace{\ell(y, \langle X, w\rangle)}_{\text{training error}} + \sum_{l=1}^{L} \rho_l(\theta_l \odot w) \Big\}$$

generalizes well to unseen data

Idea: Find θ such that $\hat{w}(\theta)$ minimizes the validation error $\mathcal{E}(\hat{w}(\theta)) = \ell(y^{\mathrm{val}}, \langle X^{\mathrm{val}}, \hat{w}(\theta)\rangle)$
$\Rightarrow$ type of continuous cross-validation

Bilevel Problem

$$\min_{\theta \in \Theta} \mathcal{E}(\hat{w}(\theta)) \qquad \text{s.t.} \qquad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

Exact Problem

$$\min_{\theta\in\Theta} \left\{\mathcal{U}(\theta) \triangleq \mathcal{E}(\hat{w}(\theta))\right\}$$

$$\text{s.t.} \quad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

$\hat{w}(\theta)$ without closed form

Bilevel Framework

Exact Problem

$$\min_{\theta \in \Theta} \left\{ \mathcal{U}(\theta) \triangleq \mathcal{E}(\hat{w}(\theta)) \right\}$$

$$\text{s.t.} \quad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w; \theta)$$

$\hat{w}(\theta)$ without closed form

Approximate Problem

$$\min_{\theta \in \Theta} \left\{ \mathcal{U}^{(k)}(\theta) \triangleq \mathcal{E}(w^{(k)}(\theta)) \right\}$$

$$\text{s.t.} \quad \begin{array}{l} w^{(0)}(\theta) \text{ chosen arbitrarily} \\ \text{for } i = 0, \dots, k-1 \\ \lfloor \; w^{(i+1)}(\theta) = \mathcal{A}(w^{(i)}(\theta)) \\ w^{(k)}(\theta) \to \hat{w}(\theta) \end{array}$$

$\mathcal{U}^{(k)}$ smooth if $\mathcal{A}$ smooth

$\Downarrow$

choice of $\mathcal{A}$ discussed next

Exact Problem

$$\min_{\theta\in\Theta} \left\{\mathcal{U}(\theta) \triangleq \mathcal{E}(\hat{w}(\theta))\right\}$$

$$\text{s.t.} \quad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

$\hat{w}(\theta)$ without closed form

Approximate Problem

$$\min_{\theta\in\Theta} \left\{\mathcal{U}^{(k)}(\theta) \triangleq \mathcal{E}(w^{(k)}(\theta))\right\}$$

s.t.
- $w^{(0)}(\theta)$ chosen arbitrarily
- for $i = 0, \dots, k-1$
  - $w^{(i+1)}(\theta) = \mathcal{A}(w^{(i)}(\theta))$
- $w^{(k)}(\theta) \to \hat{w}(\theta)$

$\mathcal{U}^{(k)}$ smooth if $\mathcal{A}$ smooth

$\theta^{(0)}$ chosen arbitrarily
for $n = 0, 1, \dots$
- $w^{(0)}(\theta^{(n)})$ chosen arbitrarily
- for $i = 0, \dots, k-1$ (inner algorithm)
  - $w^{(i+1)}(\theta^{(n)}) = \mathcal{A}(w^{(i)}(\theta^{(n)}))$
- $\theta^{(n+1)} = \mathrm{Proj}_\Theta(\theta^{(n)} - \gamma\nabla\mathcal{U}^{(k)}(\theta^{(n)}))$ where $\mathcal{U}^{(k)}(\theta^{(n)}) = \mathcal{E}(w^{(k)}(\theta^{(n)}))$

Algorithmic Solution

Optimization problem

$$\underset{w\in\mathbb{R}^P}{\text{minimize}} \;\; \underbrace{\frac{1}{2}\|y - Xw\|_2^2}_{f(w)\text{ differentiable}} + \underbrace{\lambda \sum_{l=1}^{L} \|\theta_l \odot w\|_2}_{g(A_\theta w)\text{ non differentiable}}$$

where $A_\theta : w \in \mathbb{R}^P \mapsto (\theta_1 \odot w, \ldots, \theta_L \odot w) \in \mathbb{R}^{P\times L}$

Forward-backward algorithm [Combettes and Wajs (2005)]

$$\left\{\begin{array}{l} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; w^{(i+1)}(\theta) = \operatorname{prox}_{\beta g \circ A_\theta}\left(w^{(i)}(\theta) - \beta \nabla f(w^{(i)}(\theta))\right) \right. \end{array}\right.$$

proximity operator (prox) *?* $\longrightarrow$ *see next slide*

Group Lasso solver $\mathcal{A}$

Optimization problem

$$\underset{w\in\mathbb{R}^P}{\text{minimize}} \quad \underbrace{\frac{1}{2}\|y - Xw\|_2^2}_{f(w)\ \text{differentiable}} + \underbrace{\lambda\sum_{l=1}^{L}\|\theta_l \odot w\|_2}_{g(A_\theta w)\ \text{non differentiable}}$$

where $A_\theta : w \in \mathbb{R}^P \mapsto (\theta_1 \odot w, \ldots, \theta_L \odot w) \in \mathbb{R}^{P\times L}$

Forward-backward algorithm [Combettes and Wajs (2005)]

$$\left\{\begin{array}{l} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; w^{(i+1)}(\theta) = \mathrm{prox}_{\beta g\circ A_\theta}\big(w^{(i)}(\theta) - \beta\nabla f(w^{(i)}(\theta))\big) \right. \end{array}\right.$$

proximity operator (prox) *?* $\longrightarrow$ *see next slide*

Proximity operator

In the 1960s, [Moreau (1962)] proposed an extension of the notion of projection operator to any convex function h, leading to the so-called proximity operator

$$\begin{aligned}\mathrm{Proj}_{\mathcal{C}}(v) &= \operatorname*{argmin}_{w \in \mathcal{C}} \frac{1}{2}\|w - v\|_2^2 \\ &= \operatorname*{argmin}_{w \in \mathbb{R}^P} \imath_{\mathcal{C}}(w) + \frac{1}{2}\|w - v\|_2^2 \quad \text{where} \quad \imath_{\mathcal{C}}(w) = \begin{cases} 0 & \text{if } w \in \mathcal{C} \\ +\infty & \text{otherwise} \end{cases}\end{aligned}$$

$$\mathrm{prox}_h(v) = \operatorname*{argmin}_{w \in \mathbb{R}^P} h(w) + \frac{1}{2}\|w - v\|_2^2$$

Group Lasso solver $\mathcal{A}$

Optimization problem

$$\underset{w \in \mathbb{R}^P}{\text{minimize}} \; \underbrace{\frac{1}{2}\|y - Xw\|_2^2}_{f(w)\text{ differentiable}} + \underbrace{\lambda \sum_{l=1}^{L} \|\theta_l \odot w\|_2}_{g(A_\theta w)\text{ non differentiable}}$$

where $A_\theta : w \in \mathbb{R}^P \mapsto (\theta_1 \odot w, \ldots, \theta_L \odot w) \in \mathbb{R}^{P \times L}$

Forward-backward algorithm [Combettes and Wajs (2005)]

$$\left\{ \begin{array}{l} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; w^{(i+1)}(\theta) = \mathrm{prox}_{\beta g \circ A_\theta}\left(w^{(i)}(\theta) - \beta \nabla f(w^{(i)}(\theta))\right) \right. \end{array} \right.$$

generalization of projected gradient descent

projection $\longrightarrow$ proximity operator

$\mathrm{prox}_{\beta g \circ A_\theta}(v) = \underset{w \in \mathbb{R}^P}{\mathrm{argmin}} \; \beta g(A_\theta w) + \frac{1}{2}\|w - v\|_2^2$ ⚠ without closed form

Optimization problem

$$\underset{w \in \mathbb{R}^P}{\text{minimize}} \; \underbrace{\frac{1}{2}\|y - Xw\|_2^2}_{f(w) \text{ differentiable}} + \underbrace{\lambda \sum_{l=1}^{L} \|\theta_l \odot w\|_2}_{g(A_\theta w) \text{ non differentiable}}$$

where $A_\theta : w \in \mathbb{R}^P \mapsto (\theta_1 \odot w, \dots, \theta_L \odot w) \in \mathbb{R}^{P \times L}$

Forward-backward algorithm [Combettes and Wajs (2005)]

$$\left\{ \begin{array}{l} \text{for } i = 0, \dots, k-1 \\ \quad w^{(i+1)}(\theta) = \operatorname{prox}_{\beta g \circ A_\theta}\left(w^{(i)}(\theta) - \beta \nabla f(w^{(i)}(\theta))\right) \end{array} \right.$$

generalization of projected gradient descent

projection $\longrightarrow$ *proximity operator*

$$\operatorname{prox}_{\beta g \circ A_\theta}(v) = \underset{w \in \mathbb{R}^P}{\operatorname{argmin}} \; \beta g(A_\theta w) + \frac{1}{2}\|w - v\|_2^2$$ ⚠ without closed form

Duality in convex optimization

The ideas of duality and transforms are ubiquitous in mathematics

- Harmonics analysis $\to$ Fourier transform
- Convex analysis $\to$ Fenchel conjugate: $h^*(x) = \sup_w \langle w, x\rangle - h(w)$

[Rockafellar (1970)]

Example: $h\colon w \mapsto \|w\|_2 \implies h^*\colon x \mapsto \imath_{\mathcal{B}(1)}(x) = \begin{cases} 0 & \text{if } \|x\|_2 \leq 1 \\ +\infty & \text{otherwise} \end{cases}$

Duality in convex optimization

The ideas of duality and transforms are ubiquitous in mathematics

- Harmonics analysis $\rightarrow$ Fourier transform
- Convex analysis $\rightarrow$ Fenchel conjugate: $h^*(x) = \sup_w \langle w, x \rangle - h(w)$

[Rockafellar (1970)]

Example: $h\colon w \mapsto \lambda \|w\|_2 \quad \Rightarrow \quad h^*: x \mapsto \imath_{\mathcal{B}(\lambda)}(x) = \begin{cases} 0 & \text{if } \|x\|_2 \leq \lambda \\ +\infty & \text{otherwise} \end{cases}$

The ideas of duality and transforms are ubiquitous in mathematics

- Harmonics analysis $\rightarrow$ Fourier transform
- Convex analysis $\rightarrow$ Fenchel conjugate: $h^*(x) = \sup_w \langle w, x\rangle - h(w)$

[Rockafellar (1970)]

Primal problem $\longleftrightarrow$ **Dual problem**

$$\underset{w\in\mathbb{R}^P}{\text{minimize}}\ f(w) + g(A_\theta w)$$

$$A_\theta : \mathbb{R}^P \rightarrow \mathbb{R}^{P\times L}$$

$$g(v_1 \ldots v_L) = \sum_{l=1}^{L} \underbrace{\lambda\|v_l\|_2}_{\text{norm}}$$

$$\underset{u\in\mathbb{R}^{P\times L}}{\text{minimize}}\ f^*(-A_\theta^\top u) + g^*(u)$$

$$A_\theta^\top : \mathbb{R}^{P\times L} \rightarrow \mathbb{R}^P$$

$$g^*(u_1 \ldots u_L) = \sum_{l=1}^{L} \underbrace{\imath_{\mathcal{B}(\lambda)}(u_l)}_{\substack{\text{indicator}\\ \text{dual norm ball}}}$$

Link

$$w = \nabla f^*(-A_\theta^\top u)$$

$\mathrm{prox}_{g\circ A_\theta}$ without closed form $\Rightarrow$ solve dual problem to move A_θ in smooth part

Dual problem

$$\underset{u\in\mathbb{R}^{P\times L}}{\text{minimize}} \; \underbrace{f^*(-A_\theta^\top u)}_{\text{differentiable}} + \underbrace{g^*(u)}_{\text{non differentiable}}$$

Dual forward-backward algorithm

$$\begin{cases} \text{for } i = 0, \ldots, k-1 \\ \quad \big\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}_{\beta g^*}\left(u^{(i)}(\theta) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\right) \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta^* u^{(k)}(\theta)) \qquad \textbf{(link)} \end{cases}$$

Dual problem

$$\underset{u\in\mathbb{R}^{P\times L}}{\text{minimize}}\ \underbrace{f^*(-A_\theta^\top u)}_{\text{differentiable}} + \underbrace{g^*(u)}_{\text{non differentiable}}$$

Dual forward-backward algorithm

$$\begin{cases} \text{for } i = 0,\ldots,k-1 \\ \quad \left\lfloor\ u^{(i+1)}(\theta) = \operatorname{prox}_{\beta g^*}\left(u^{(i)}(\theta) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\right) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta^* u^{(k)}(\theta)) \qquad \textbf{(link)} \end{cases}$$

where the proximal operator reads:

$$\begin{aligned} \operatorname{prox}_{\beta g^*}(v) &= \underset{u\in\mathbb{R}^{P\times L}}{\operatorname{argmin}}\ \beta g^*(u) + \frac{1}{2}\|u-v\|^2 \\ &= \underset{u\in\mathbb{R}^{P\times L}}{\operatorname{argmin}}\ \beta \sum_{l=1}^{L} \imath_{\mathcal{B}(\lambda)}(u_l) + \frac{1}{2}\|u-v\|^2 \\ &= \operatorname{Proj}_{\mathcal{B}(\lambda)^L}(v) \end{aligned}$$

⚠ not differentiable

$\theta^{(0)}$ chosen arbitrarily
for $n = 0, 1, \ldots$
  $w^{(0)}(\theta^{(n)})$ chosen arbitrarily
  for $i = 0, \ldots, k-1$ (inner algorithm = dual forward-backward)
    $w^{(i+1)}(\theta^{(n)}) = \mathcal{A}(w^{(i)}(\theta^{(n)}))$
  $\theta^{(n+1)} = \mathrm{Proj}_\Theta(\theta^{(n)} - \gamma \nabla \mathcal{U}^{(k)}(\theta^{(n)}))$ where $\mathcal{U}^{(k)}(\theta^{(n)}) = \mathcal{E}(w^{(k)}(\theta^{(n)}))$

We want a differentiable dual forward-backward algorithm
because it inside a bilevel algorithm !

Group Lasso solver $\mathcal{A}$: dual approach

Dual forward-backward algorithm

$$\begin{cases} \text{for } i = 0, \ldots, k-1 \\ \quad \left\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}_{\beta g^*}\big(\quad u^{(i)}(\theta) \; + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\big) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{cases}$$

where

$$\begin{aligned} \mathrm{prox}_{\beta g^*}(v) &= \underset{u \in \mathbb{R}^{P \times L}}{\mathrm{argmin}}\, \beta g^*(u) + \frac{1}{2}\|u - v\|^2 \\ &= \underset{u \in \mathbb{R}^{P \times L}}{\mathrm{argmin}}\, \beta g^*(u) + \frac{1}{2}\|u\|^2 - \langle u, v \rangle + \mathrm{cst} \end{aligned}$$

Group Lasso solver $\mathcal{A}$: dual approach

Dual forward-backward algorithm

$$\begin{cases} \text{for } i = 0, \ldots, k-1 \\ \quad \left\lfloor \; u^{(i+1)}(\theta) = \operatorname{prox}_{\beta g^*}\left(\quad u^{(i)}(\theta) \;\; + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\right) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{cases}$$

where

$$\operatorname{prox}_{\beta g^*}(v) = \underset{u \in \mathbb{R}^{P \times L}}{\operatorname{argmin}} \, \beta g^*(u) + \frac{1}{2}\|u\|^2 - \langle u, v \rangle + \text{cst}$$

Dual forward-backward algorithm with Bregman distances [Bauschke et al. (2016)]

$$\begin{cases} \text{for } i = 0, \dots, k-1 \\ \left\lfloor \; u^{(i+1)}(\theta) = \operatorname{prox}^{\Phi}_{\beta g^*}\big(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\big) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{cases}$$

where the Bregman proximal operator associated to Φ :

$$\operatorname{prox}^{\Phi}_{\beta g^*}(v) = \underset{u \in \mathbb{R}^{P \times L}}{\operatorname{argmin}}\, \beta g^*(u) + \Phi(u) - \langle u, v \rangle$$

Choice of Φ to smooth the updates

Dual forward-backward algorithm with Bregman distances

$$\left\{\begin{array}{l} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}^{\Phi}_{\beta g^*}\big(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\big) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{array}\right.$$

where

$$\mathrm{prox}^{\Phi}_{\beta g^*}(v) = \underset{u\in\mathbb{R}^{P\times L}}{\mathrm{argmin}}\, \beta g^*(u) + \Phi(u) - \langle u, v\rangle$$

Choice of Φ to smooth the updates

Dual forward-backward algorithm with Bregman distances

$$\left\{\begin{array}{l} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}^{\Phi}_{\beta g^*}\big(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\big) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{array}\right.$$

where

$$\mathrm{prox}^{\Phi}_{\beta g^*}(v) = \underset{u\in\mathbb{R}^{P\times L}}{\mathrm{argmin}} \sum_{l=1}^{L} \imath_{\mathcal{B}(\lambda)}(u_l) + \Phi(u) - \langle u, v\rangle$$

Choice of Φ to smooth the updates

Dual forward-backward algorithm with Bregman distances

$$\begin{cases} \text{for } i = 0, \ldots, k-1 \\ \left\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}^{\Phi}_{\beta g^*}\big(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\big) \right. \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{cases}$$

where

$$\mathrm{prox}^{\Phi}_{\beta g^*}(v) = \underset{u \in \mathbb{R}^{P \times L}}{\mathrm{argmin}} \sum_{l=1}^{L} \big(\imath_{\mathcal{B}(\lambda)}(u_l) + \phi(u_l) - \langle u_l, v_l \rangle\big)$$

for $\Phi(u) = \sum_{l=1}^{L} \phi(u_l)$

Choice of Φ to smooth the updates

Dual forward-backward algorithm with Bregman distances

$$\begin{cases} \text{for } i = 0, \ldots, k-1 \\ \lfloor \; u^{(i+1)}(\theta) = \operatorname{prox}^{\Phi}_{\beta g^*}\left(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\right) \\ w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)). \end{cases}$$

where

$$\operatorname{prox}^{\Phi}_{\beta g^*}(v) = \underset{u \in \mathbb{R}^{P \times L}}{\operatorname{argmin}} \sum_{l=1}^{L} \left(\imath_{\mathcal{B}(\lambda)}(u_l) - \sqrt{\lambda^2 - \|u_l\|^2} - \langle u_l, v_l \rangle \right)$$

for $\phi(u_l) = -\sqrt{\lambda^2 - \|u_l\|^2} \Rightarrow \operatorname{dom}\phi = \mathcal{B}(\lambda)$

$\Rightarrow \imath_{\mathcal{B}(\lambda)}(u_l)$ always equal to 0 !

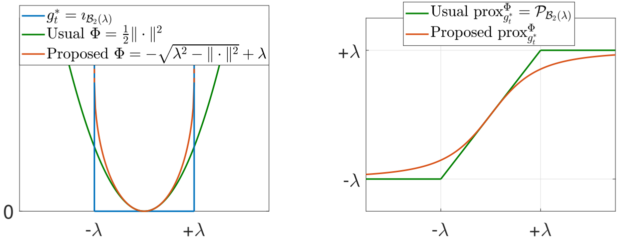
⚠ **trick for a differentiable algorithm**

Dual forward-backward algorithm with Bregman distances

$$
\begin{cases}
\text{for } i = 0, \ldots, k-1 \\
\quad \left\lfloor \; u^{(i+1)}(\theta) = \mathrm{prox}^{\Phi}_{\beta g^*}\left(\nabla\Phi(u^{(i)}(\theta)) + \beta A_\theta \nabla f^*(-A_\theta^\top u^{(i)}(\theta))\right) \right. \\
w^{(k)}(\theta) = \nabla f^*(-A_\theta u^{(k)}(\theta)).
\end{cases}
$$

where

$$
\mathrm{prox}^{\Phi}_{\beta g^*}(v) = \left(\frac{\lambda v_l}{\sqrt{1 + \|v_l\|_2^2}} \right)_{l=1,\ldots,L}
$$



Convergence $w^{(k)}(\theta) \to \hat{w}(\theta)$

Theorem 1: For every $\theta \in \Theta$, $\|w^{(k)}(\theta) - \hat{w}(\theta)\|^2 \leq \dfrac{\text{Const}}{k}$

Convergence "$\mathcal{U}^{(k)}(\theta) \to \mathcal{U}(\theta)$"

Theorem 2: Assume that Θ is a non-empty compact subset of $\mathbb{R}_+^{P\times L}$. If the iterates $\{\mathrm{w}^{(k)}(\theta)\}_{k\in\mathbb{N}}$ converge to $\hat{\mathrm{w}}(\theta)$ uniformly in Θ when $k \to +\infty$, then

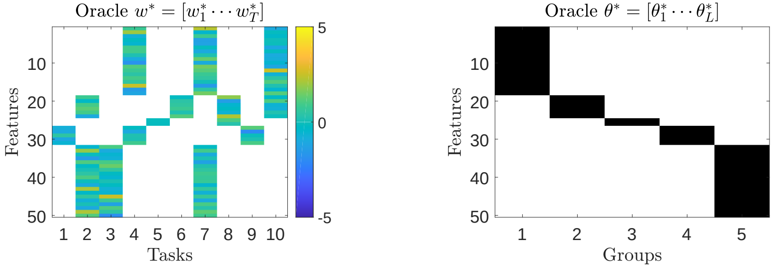
$$\inf_{\theta\in\Theta} \mathcal{U}^{(k)}(\theta) \underset{k\to+\infty}{\longrightarrow} \inf_{\theta\in\Theta} \mathcal{U}(\theta) \qquad \text{and} \qquad \underset{\theta\in\Theta}{\operatorname{argmin}}\ \mathcal{U}^{(k)}(\theta) \underset{k\to+\infty}{\longrightarrow} \underset{\theta\in\Theta}{\operatorname{argmin}}\ \mathcal{U}(\theta)$$

Reminder : $$\begin{cases} \mathcal{U}(\theta) & = \mathcal{E}(\hat{w}(\theta)) \\ \mathcal{U}^{(k)}(\theta) & = \mathcal{E}(\hat{w}^{(k)}(\theta)) \end{cases}$$

Numerical Experiments

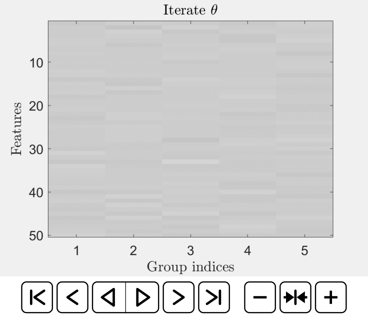
Setting: $T = 500$ tasks, $N = 25$ noisy observations, $P = 50$ parameters.

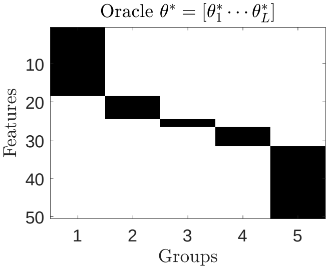
Goal: Estimate and group the parameters



$$y_t = X_t w_t^* + \epsilon_t \quad \text{where} \quad \epsilon_t \sim \mathcal{N}(0, \sigma = 0.1)$$

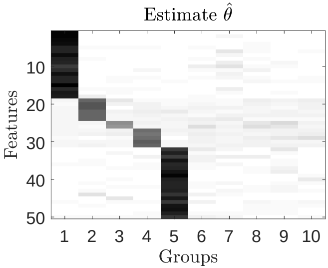
Result

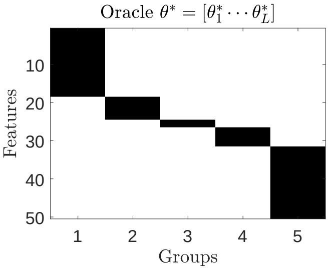




Recover the correct groups ! (just different ordering)

When the number of groups is unknown





Works even when the number of groups is unknown !

Conclusion

Conclusion

1. Define structured predictor with groups θ

$$\hat{w}(\theta) = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(w; \theta)$$

2. **Ideal:** find groups θ such that $\hat{w}(\theta)$ minimizes the validation error

$$\min_{\theta \in \Theta} \mathcal{E}(\hat{w}(\theta)) \quad \text{s.t.} \quad \hat{w}(\theta) = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(w; \theta)$$

3. **Practice:** solve a differentiable bilevel problem

$$\min_{\theta \in \Theta} \mathcal{E}(w^{(k)}(\theta)) \quad \text{s.t.} \quad \begin{array}{l} w^{(0)}(\theta) \text{ chosen arbitrarily} \\ \text{for } i = 0, \ldots, k-1 \\ \lfloor \; w^{(i+1)}(\theta) = \mathcal{A}(w^{(i)}(\theta)) \end{array} \qquad \text{with } \mathcal{A} \text{ differentiable}$$

What is next

1. More complex structures (overlapping, hierarchical, ...)
2. New multi-task models to transfer learning
3. Theoretical guarantees for bilevel optimization (global minima, convergence rate, ...)

Thank you

"H. H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. Mathematics of Operations Research, 2016."

"P. L Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. In SIAM Multiscale Modeling & Simulation. 2005"

"J. Frecon, S. Salzo, and M. Pontil. Bilevel learning of the group Lasso structure. In Advances in Neural Information Processing Systems (NeurIPS). 2018"

"J. Frecon, S. Salzo, and M. Pontil. Unveiling groups of related tasks in multi-task learning. In Proceedings of the International Conference on Patterns Recognition (ICPR). 2020"

"C. Higuera, K.J. Gardiner, K.J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. In PLoS ONE. 2015"

"Z. Kang, K. Grauman and F. Sha. Learning with whom to share in multi-task feature learning. In Proceedings of the International Conference on Machine Learning (ICML), 2011."

"M. Kshirsagar, E. Yang, A.C. Lozano. learning task clusters via sparsity grouped multitask learning. In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), 2017."

"C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009"

"J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. Comptes Rendus de l'Académie des Sciences de Paris. 1962."

"T. K. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: reformulations, algorithms, and multi-task learning. In SIAM Journal of Optimization. 2010"

"R.T.Rockafellar. Convex analysis. Princeton university press. 1970"

"D. Sabbagh, P. Ablin, G. Varoquaux, A. Gramfort, and D. Engemann. Manifold-regression to predict from MEG/EEG brain signals without source modeling In Advances in Neural Information Processing Systems. 2019"

"M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. In Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2006"

What is next (theoretical guarantees)

Exact problem

$$\min_{\theta\in\Theta} \left\{\mathcal{U}(\theta) \triangleq \mathcal{E}(\hat{w}(\theta))\right\}$$

s.t. $\hat{w}(\theta) = \operatorname*{argmin}_{w\in\mathcal{W}} \mathcal{L}(w;\theta)$

Approximate problem

$$\min_{\theta\in\Theta} \left\{\mathcal{U}^{(k)}(\theta) \triangleq \mathcal{E}(w^{(k)}(\theta))\right\}$$

s.t. $w^{(0)}(\theta)$ chosen arbitrarily

for $i = 0, \ldots, k-1$

$\lfloor\; w^{(i+1)}(\theta) = \mathcal{A}(w^{(i)}(\theta))$

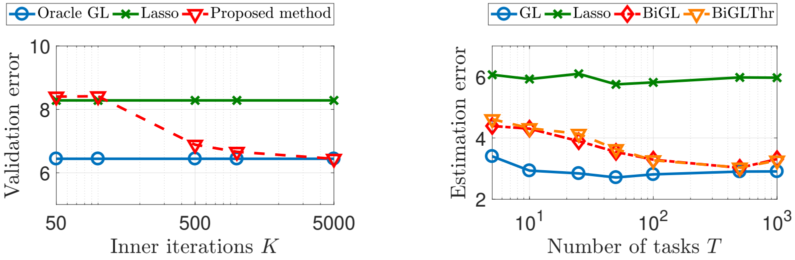
$w^{(k)}(\theta) \to \hat{w}(\theta)$

$$\theta^{(n+1)} = \mathrm{Proj}_\Theta(\theta^{(n)} - \mu\nabla\mathcal{U}^{(k)}(\theta^{(n)}))$$

- $\inf_{\theta\in\Theta} \mathcal{U}^{(k)}(\lambda) \xrightarrow[k\to+\infty]{} \inf_{\theta\in\Theta} \mathcal{U}(\lambda)$ ✓
- $\operatorname*{argmin}_{\theta\in\Theta} \mathcal{U}^{(k)}(\lambda) \xrightarrow[k\to+\infty]{} \operatorname*{argmin}_{\theta\in\Theta} \mathcal{U}(\lambda)$ ✓
- Efficient computation of $\nabla\mathcal{U}^{(k)}$ ✓

- Impact of warm-restart on $w^{(0)}$?
- $\lim_{k\to\infty} \nabla\mathcal{U}^{(k)}(\theta) \in \partial\mathcal{U}(\theta)$?
- $\lim_{n\to\infty} \theta^{(n)} \in \partial\mathcal{U}^{-1}(0)$?

Reminder : $[w_1^{(k)}(\theta) \cdots w_T^{(k)}(\theta)] \to [\hat{w}_1(\theta) \cdots \hat{w}_T(\theta)]$



(GL) group Lasso with oracle groups
(Lasso) Lasso
(BiGL) proposed method

Convergence of the upper iterates

Convergence to a stationary point

Theorem 3: For $\bar{n}$ uniformly sampled in $\{1, \ldots, n_{\max}\}$:

$$\mathbb{E}\left[\|G_\gamma(\theta^{(\bar{n})})\|^2\right] \leq \frac{\text{Const}}{n_{\max}},$$

where G_γ with step-size γ

$$G_\gamma(\theta) = \frac{1}{\gamma}\left(\theta - \mathcal{P}_\Theta(\theta - \gamma \nabla \mathcal{U}^{(k)}(\theta)\right)$$

Intuition : Without the projection, $G_\gamma(\theta) = \nabla \mathcal{U}^{(k)}(\theta)$

$$\begin{cases} u^{(0)}(\theta) \in \mathcal{H} \\ \text{for } i = 0, \dots, k-1 \\ \left\lfloor \; u^{(i+1)}(\theta) = \mathcal{A}(u^{(i)}(\theta), \theta) \right. \\ w^{(k)}(\theta) = \mathcal{B}(u^{(k)}(\theta), \theta), \end{cases} \tag{1}$$

we get

$$\nabla \mathcal{U}^{(k)}(\theta) = (u^{(k)})'(\theta)^\top \partial_1 \mathcal{B}(u^{(k)}(\theta), \theta)^\top \nabla C(w^{(k)}(\theta)) + \partial_2 \mathcal{B}(u^{(k)}(\theta), \theta)^\top \nabla C(w^{(k)}(\theta)). \tag{2}$$

Moreover, using the updating rule for $u^{(i)}(\theta)$ in (1) we have

$$(u^{(i+1)})'(\theta) = \partial_1 \mathcal{A}(u^{(i)}(\theta), \theta)(u^{(i)})'(\theta) + \partial_2 \mathcal{A}(u^{(i)}(\theta), \theta). \tag{3}$$

Setting $A_1^{(i)}(\theta) = \partial_1 \mathcal{A}(u^{(i)}(\theta), \theta)$ and $A_2^{(i)}(\theta) = \partial_2 \mathcal{A}(u^{(i)}(\theta), \theta)$, we have

$$(u^{(i+1)})'(\theta)^\top = (u^{(i)})'(\theta)^\top A_1^{(i)}(\theta)^\top + A_2^{(i)}(\theta)^\top. \tag{4}$$

Then, by combining the two equations above we have

$$\begin{aligned}
\nabla\mathcal{U}^{(k)}(\theta) &= (u^{(k)})'(\theta)^\top \partial_1\mathcal{B}(u^{(k)}(\theta),\theta)^\top \nabla C(w^{(k)}(\theta)) + \partial_2\mathcal{B}(u^{(k)}(\theta),\theta)^\top \nabla C(w^{(k)}(\theta)) \\
&= (u^{(k-1)})'(\theta)^\top A_1^{(k-1)}(\theta)^\top \underbrace{\partial_1\mathcal{B}(u^{(k)}(\theta),\theta)^\top \nabla C(w^{(k)}(\theta))}_{a_Q} \\
&\quad + A_2^{(k-1)}(\theta)^\top \underbrace{\partial_1\mathcal{B}(u^{(k)}(\theta),\theta)^\top \nabla C(w^{(k)}(\theta))}_{a_k} + \underbrace{\partial_2\mathcal{B}(u^{(k)}(\theta),\theta)^\top \nabla C(w^{(k)}(\theta))}_{b_k} \\
&= (u^{(k-1)})'(\theta)^\top \underbrace{A_1^{(k-1)}(\theta)^\top a_k}_{a_{k-1}} + \underbrace{A_2^{(k-1)}(\theta)^\top a_k + b_k}_{b_{k-1}} \\
&= (u^{(k-2)})'(\theta)^\top \underbrace{A_1^{(k-2)}(\theta)^\top a_{k-1}}_{a_{k-2}} + \underbrace{A_2^{(k-2)}(\theta)^\top a_{k-1} + b_{k-1}}_{b_{k-2}} \\
&= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&= \underbrace{A_2^{(0)}(\theta)^\top a_1 + b_1}_{b_0},
\end{aligned}$$

where in the last line we used that $u^{(0)}(\theta)$ is constant.

State-of-the-art: joint optimization [Kang et al. (2011), Kshirsagar et al. (2017).]

$$(\hat{w}, \hat{\theta}) = \operatorname*{argmin}_{w,\theta\in\Theta} \left\{ \mathcal{L}(w;\theta) \triangleq \ell(y, \langle X, w\rangle) + \sum_{l=1}^{L} \rho_l(\theta_l \odot w) \right\}$$

issues: some trivial undesired minima
unclear interpretation of the solution

Proposed method: bilevel optimization [Frecon et al. (2018), Frecon et al. (2020).]

$$\min_{\theta\in\Theta} \mathcal{E}(\hat{w}(\theta)) \qquad \text{s.t.} \qquad \hat{w}(\theta) = \operatorname*{argmin}_{w} \mathcal{L}(w;\theta)$$

idea: find θ such that $\hat{w}(\theta)$ generalizes well to unseen data
$\rightarrow$ choose $\mathcal{E}$ as the validation error