A fast algorithm for structured gene selection

MLSB 2010

Sofia Mosci
DISI, Università degli Studi di Genova

Joint work with
A. Verri(1), S. Villa(1), and L. Rosasco(2)
1 - Universita' di Genova 2-IIT-MIT
Gene selection problem

extracting a predictive model depending on a small subset of genes

many variable selection algorithms are available (filters wrappers and embedded)

- low accuracy
- low stability
- low interpretability
Motivation

Gene selection problem

extracting a predictive model depending on a small subset of genes

many variable selection algorithms are available (filters wrappers and embedded)

- low accuracy
- low stability
- low interpretability
Motivation

Strong prior is often available!
Genes must be selected according to groups defined a priori.

Examples of groups:
- GO
- KEGG
- ad hoc grouping

Group lasso

References:
Lanckriet et al.’04, Meier et al. ’06, Yuan-Lin ’06, Bach ’08,...

Group lasso drawback: groups must be a partition of the genes
Motivation

Group lasso with overlap (Jacob, Obozinski and Vert ’09)

Genes must be selected group-wise according to groups defined a priori. Like group lasso but groups may overlap.

Advantages:
- Higher stability
- Higher accuracy
- Higher interpretability

Disadvantages:
- Implementability
Motivation

Goal

to develop a scalable approach to group lasso with overlap

Plan:

- Proximal methods for Sparsity based regularization
- Group lasso with overlap: the initial approach
- Group lasso with overlap: our projection algorithm
- Experiments
General sparsity prior: variables are organized in separate, nested or possibly overlapping groups.

Given a training set \((x_i, y_i)_{i=1}^n\), with \(x_i \in \mathbb{R}^d\), consider

\[
\arg\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \| X \beta - y \|^2 + \frac{2\tau}{n} \Omega(\beta) \right\}
\]

where

- \([X]_{i,j} = (x_i)_j\)
- \(\Omega : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}\), encodes the sparsity prior, and is convex and one-homogeneous (\(\Omega(\lambda \beta) = \lambda \Omega(\beta), \forall \beta \in \mathbb{R}^d\) and \(\lambda \in \mathbb{R}^+\)).
A Proximal Algorithm

Require: $\tau, \sigma > 0$
Initialize: $\beta^0 = 0$
while convergence not reached do
    $p := p + 1$
    $\beta^p = \text{prox}_{\frac{\tau}{\sigma} \Omega} \left( \beta^{p-1} - \frac{1}{n\sigma} X^T (X \beta^{p-1} - y) \right)$
end while
return $\beta^p$

References:
- Lions-Mercier ('79), Passty ('76), Tseng (90s), Chen-Rockafellar ('89), Eckstein ('89), Combettes-Wajs ('05)
- Duchi and Singer '09, Jenatton et al. '10, Mosci et al. '10 for machine learning
A Proximal Algorithm

Require: \( \tau, \sigma > 0 \)
Initialize: \( \beta^0 = 0 \)

while convergence not reached do

\[ p := p + 1 \]

\[ \beta^p = \text{prox}_{\frac{\tau}{\sigma} \Omega} \left( \beta^{p-1} - \frac{1}{n\sigma} X^T (X \beta^{p-1} - y) \right) \]

end while

return \( \beta^p \)

References:

- Lions-Mercier ('79), Passty ('76), Tseng (90s), Chen-Rockafellar ('89), Eckstein ('89), Combettes-Wajs ('05)
- Duchi and Singer '09, Jenatton et al. '10, Mosci et al. '10 for machine learning
A Proximal Algorithm

Require: \( \tau, \sigma > 0 \)
Initialize: \( \beta^0 = 0 \)

while convergence not reached do

\[ p := p + 1 \]

\[ \beta^p = \operatorname{prox}_{\frac{\tau}{\sigma}} \Omega \left( \beta^{p-1} - \frac{1}{n\sigma} X^T (X \beta^{p-1} - y) \right) \]

end while

return \( \beta^p \)

References:

- Lions-Mercier ('79), Passty ('76), Tseng (90s), Chen-Rockafellar ('89), Eckstein ('89), Combettes-Wajs ('05)
- Duchi and Singer '09, Jenatton et al. '10, Mosci et al. '10 for machine learning
Iterative soft-thresholding for the lasso

**Prior:** the relevant variables are a subset of the total variables

\[
\arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \| X \beta - y \|^2 + 2\tau \| \beta \|_1
\]

**Require:** \( \tau, \sigma > 0 \)

**Initialize:** \( \beta^0 = 0 \)

**while** convergence not reached **do**

\[ p := p + 1 \]

\[
\beta^p = S_{\frac{\tau}{\sigma}} \left( \beta^{p-1} - \frac{1}{n\sigma} X^T (X \beta^{p-1} - y) \right)
\]

**end while**

**return** \( \beta^p \)

where \( S \) is the soft-thresholding operator: \( S_{\lambda}(\beta^j):=(|\beta^j| - \lambda) + \text{sign}(\beta^j) \)

**References:**
Daubechies et al. ‘04, Combettes ‘05, Figuereido et al. ‘07
Prior: the relevant variables are union of a subset of the $B$ groups given a priori, $\{G_r\}_{r=1}^B$, that make a block partition of $\{1, \ldots, d\}$

$$\argmin_{\beta \in \mathbb{R}^d} \frac{1}{n} \|X\beta - y\|^2 + 2\tau \sum_{r=1}^M \sqrt{\sum_{j \in G_r} \beta_j^2} \Omega$$

$$\beta^p = \tilde{S}_{\frac{\tau}{\sigma}} \left( \beta^{p-1} - \frac{1}{n\sigma} X^T (X\beta^{p-1} - y) \right)$$

where $\tilde{S}$ is the group-wise soft-thresholding operator:

$$\tilde{S}_\lambda(\beta_k) = (\|\beta_k\|_k - \lambda) + \frac{\beta_k}{\|\beta_k\|_k}$$
**Prior:** the relevant variables are the union of a small subset of the $B$ groups given a priori, $\mathcal{G} = \{G_r\}_{r=1}^B$ with $G_r \subset \{1, \ldots, d\}$

Like Group Lasso but groups may overlap

$$\arg\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{n} \|X\beta - y\|^2 + 2\tau \Omega_{\text{overlap}}^G(\beta) \right\},$$

$$\Omega_{\text{overlap}}^G(\beta) = \inf_{(v_1, \ldots, v_M), v_r \in \mathbb{R}^d, \supp(v_r) \subset G_r, \sum_{r=1}^M v_r = \beta} \sum_{r=1}^M \|v_r\|.$$

**Reference:**
Jacob, Obozinski and Vert, *Group Lasso with Overlap and Graph Lasso*, ICML 2009
Group Lasso with overlap: the replication approach

A simple implementation is obtained by replicating variables belonging to more than one group, and using any algorithm for standard group lasso (e.g iterative group-wise soft-thresholding).

**Drawback**: as the degree of overlap increases the dimensionality increases and the computational burden may become very high!
Group Lasso with overlap: projection algorithm

\[ \text{prox}_{\tau \Omega^g_{\text{overlap}}} = I - \pi_{\tau} K \]

\( K \) is the intersection of cylinders centered in a coordinate subspace.
Group Lasso with overlap: projection algorithm

Sofia Mosci

A fast algorithm for structured gene selection
Group Lasso with overlap: projection algorithm

\[ \text{prox}_{\tau_{\Omega G}^{\text{overlap}}} = I - \pi_{\tau K} \]

\( K \) is the intersection of cylinders centered in a coordinate subspace.

Only a (small) subset of the cylinders are active.

For a given \( \beta \in \mathbb{R}^d \), the projection onto \( \tau K \) is given by

\[
\argmin_{\|v\|_G \leq \tau, \, \forall G \in \hat{G}} \|v - \beta\|^2
\]

where \( \hat{G} := \{ G \in \mathcal{G}, \|\beta\|_G > \tau \} \) is the set of active groups.
Group Lasso with overlap: projection algorithm

A fast algorithm for structured gene selection
Group Lasso with overlap: projection algorithm

A fast algorithm for structured gene selection
Group Lasso with overlap: projection algorithm
Group Lasso with overlap: projection algorithm
Group Lasso with overlap: projection algorithm

\[ \text{prox}_{\tau \Omega_{\text{overlap}}} = I - \pi_{\tau K} \]

\(K\) is the intersection of cylinders centered in a coordinate subspace.

Only a (small) subset of the cylinders are active.

For a given \(\beta \in \mathbb{R}^d\), the projection onto \(\tau K\) is given by

\[
\arg\min_{v} \|v - \beta\|^2 \\
\text{s.t.} \quad v \in \mathbb{R}^d, \|v\|_G \leq \tau \text{ per } G \in \hat{G}.
\]

where \(\hat{G} := \{G \in G, \|\beta\|_G > \tau\}\) is the set of active groups.

The projection can be computed by solving the dual problem in \(\mathbb{R}^{\hat{B}}\)

\[
\lambda^* = \arg\max_{\lambda \in \mathbb{R}^+_{\hat{B}}} \sum_{j=1}^{d} \frac{-w^2_j}{1+\sum_{r=1}^{\hat{B}} 1(j \in \hat{G}_r)\lambda_r} - \sum_{r=1}^{\hat{B}} \lambda_r \tau^2,
\]
Group Lasso with overlap: projection algorithm

**Gradient step**

\[ w = \beta^{p-1} - \frac{1}{\sigma} X^T (X \beta^{p-1} - y) \]

**Projection**

- find the set of active groups \( \hat{G} := \{ \hat{G}_1, \ldots, \hat{G}_{\hat{B}} \} \)
- compute \( \lambda^* \) solution of the dual problem associated to the reduced projection:

\[
\lambda^* = \arg \max_{\lambda \in \mathbb{R}^{\hat{B}}} \sum_{j=1}^{d} \frac{-w_j^2}{1 + \sum_{r=1}^{\hat{B}} 1(j \in \hat{G}_r) \lambda_r} - \sum_{r=1}^{\hat{B}} \lambda_r \tau^2, \\
\]

- \( \beta^p_j = w_j - \frac{w_j}{(1 + \sum_{r=1}^{\hat{B}} 1(j \in \hat{G}_r) \lambda^*_r)} \) for \( j = 1, \ldots, d \)

Overall convergence is still guaranteed!

For more details see the forthcoming paper:
Mosci, Verri, Villa and Rosasco, *A primal-dual algorithm for group \( \ell_1 \) regularization with overlapping groups*, NIPS 2010.
Experiments: projection vs duplication

3 relevant groups (with 20% overlap) for a total of 240 variables

for $k > 3$, $G_k$ is built by drawing 100 indices from $[1, \ldots, d]$

$n = 2400$

Running time (in seconds) vs $d$

![Graphs showing running time vs d for different overlap degrees](image-url)
### Experiments: microarray data

Microarray experiment presented in Jacob, Obozinski and Vert ’09 on breast cancer (Van de Vijver et al. ’01)

- 8141 genes
- 295 tumors
- 637 gene groups (Subramanian et al. 2005).
- 3-fold cross validation

<table>
<thead>
<tr>
<th>Replication</th>
<th>Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>loss:</td>
<td>logistic</td>
</tr>
<tr>
<td>prediction error:</td>
<td>0.36 ± 0.03</td>
</tr>
<tr>
<td># of selected pathways:</td>
<td>6, 5 and 78</td>
</tr>
<tr>
<td>computing time:</td>
<td>–</td>
</tr>
</tbody>
</table>

Frequency of selected groups for the Projection algorithm:

- **Split 1**
- **Split 2**
- **Split 3**

Sofia Mosci | A fast algorithm for structured gene selection
I have presented an iterative procedure for solving the group lasso with overlap regularization problem that

- is based on proximal methods and an ad hoc lemma
- is convergent
- is fast and can deal with large data sets

(code available at: www.disi.unige.it/person/MosciS/CODE/Prox.html)

I have not discussed:

- accelerations of the basic schemes
  - Continuation Methods (Hale, Yin and Zhang '08)
  - Adaptive Step size
  - Nesterov Method, linear → quadratic convergence! (Nesterov '83, Guler '91, Beck and Teboulle '09)
- other loss functions
Concluding Remarks

I have presented an iterative procedure for solving the group lasso with overlap regularization problem that
- is based on proximal methods and an ad hoc lemma
- is convergent
- is fast and can deal with large data sets

(code available at: www.disi.unige.it/person/MosciS/CODE/Prox.html)

I have not discussed:
- accelerations of the basic schemes
  - Continuation Methods (Hale, Yin and Zhang ’08)
  - Adaptive Step size
  - Nesterov Method, linear $\rightarrow$ quadratic convergence! (Nesterov ’83, Guler ’91, Beck and Teboulle ’09)
- other loss functions
A fast algorithm for structured gene selection
Experimental protocol

For each \((\tau, \lambda)\):

- Variable **selection** via Sparse Learning Algorithm with parameter \(\tau\) on training set
- **Regression** via Regularized Least Squares (RLS) on training set with parameter \(\lambda\) on selected variables
- Error estimation on **validation** set (hold-out or cross-validation)

**Minimization** of the validation error \(\rightarrow (\tau_{opt}, \lambda_{opt})\)

- Error estimation on **test** set
For one-homogeneous functionals we have the following result

Let $K$ denote the subdifferential of $\Omega$, $\partial \Omega(0)$, at the origin (which is a convex and closed subset of $\mathbb{R}^d$). For any $\lambda \in \mathbb{R}^+$ we let $\pi_{\lambda K} : \mathbb{R}^d \to \mathbb{R}^d$ be the projection on $\lambda K \subset \mathbb{R}^d$.

Then

$$\text{prox}_{\lambda \Omega} = (I - \pi_{\lambda K})$$