Convex optimization for learning Gene Regulatory Network

Magali Champion

Sébastien Gadat, Christine Cierco-Ayrolles et Matthieu Vignes

14 juin 2013
Introduction

Presentation of the problem
- Model
- Linear regression
- Optimization problem
- Oracle inequality

Procedure of optimization
- With P fixed
- With T fixed
- Alternate minimization

Permutation matrices
- Convex relaxation?
- Genetic algorithms

Numerical results
Objective: Recover the unknown gene network $\mathcal{G}$ for which:

- a node of $\mathcal{G}$ is one of the $p$ genes,
- an edge of $\mathcal{G}$ represents an interaction between two genes.
Introduction (statistical)

- $p$ studied genes, for which we know the expression data
- sample of size $n$

Objective: Recover the unknown gene network $G$ for which:

- a node of $G$ is one of the $p$ genes,
- an edge of $G$ represents an interaction between two genes.
The first idea consists in considering gene $G^j$ as an observation and the others genes as explanatory variables.

$$\forall 1 \leq j \leq p, \quad G^j = \sum_{1 \leq i \neq j \leq p} G^i + \varepsilon.$$
The first idea consists in considering gene $G_j$ as an observation and the others genes as explanatory variables.

$$\forall 1 \leq j \leq p, \quad G_j = \sum_{1 \leq i \neq j \leq p} \theta_{ij} G_i + \varepsilon.$$ 

$\Theta = (\theta^1, ..., \theta^p)$ is the adjacency matrix associated to the graph $G$, which support is denoted $S$. 

$$\Theta = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0.8 & 0 & 2 & 0.8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$
We can first rewrite the model as the following way:

\[ Y = X\theta + \varepsilon. \]

→ Main disadvantage: we don’t exploit the structure of the graph.
Consider the set of gaussian *Directed Acyclic Graphs.*

**Proposition**

Any adjacency matrix $\Theta$ associated to a DAG $G$ satisfies:

$$\Theta = PT^t P,$$

where $P$ and $T$ are permutation and lower-triangular matrices.

\[
T = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 \\
2 & 0.8 & 0.8 & 0 & 0
\end{pmatrix}
\]
Consider the set of gaussian Directed Acyclic Graphs.

**Proposition**

Any adjacency matrix $\Theta$ associated to a DAG $\mathcal{G}$ satisfies:

$$\Theta = PT^t P,$$

where $P$ and $T$ are permutation and lower-triangular matrices.

$\mathcal{G}_2 \rightarrow \mathcal{G}_1$ with $\theta = 0.8$ and $\mathcal{G}_2 \rightarrow \mathcal{G}_4$ with $\theta = 0.8$ and $\mathcal{G}_3 \rightarrow \mathcal{G}_4$ with $\theta = -1$.

$\mathcal{G}_1$ has $\theta = 2$ and $\mathcal{G}_3$ has $\theta = -1$.

$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Optimization problem

We aim at minimizing the negative penalized log-likelihood:

\[
(\hat{P}, \hat{T}) = \arg\min_{P \in \mathcal{P}_p(\mathbb{R}), T \in \mathcal{T}_p(\mathbb{R})} \left\{ \frac{1}{n} \| G - GPT^t P \|_F^2 + \lambda \| T \|_1 \right\},
\]

where

- \( \mathcal{P}_p(\mathbb{R}) \) is the set of permutation matrices,
- \( \mathcal{T}_p(\mathbb{R}) \) is the set of strict lower-triangular matrices,
- \( \| M \|_F = \text{Trace}(^t MM) = \sum_{i,j} |M_{ij}|^2 \),
- \( \| M \|_1 = \sum_{i,j} |M_{ij}| \).
Oracle inequality

Let $\hat{\Theta}$ an estimator of the parameter $\Theta^*$ and $R(.)$ a risk function. Oracle inequalities aim at comparing the risk of the proposed estimator with the risk of the "oracle", defined as the unknown parameter which minimizes the risk.

**Theorem (Oracle inequality)**

$$R(\hat{\Theta}) \leq \inf_{\Theta} \{ R(\Theta) + \text{ residual term} \}.$$
Oracle inequality

Assumption $Re(s, c_0)$: for some integer $s \in \{1, \ldots, p\}$, and $c_0 \geq 0$, the following condition holds:

$$
\kappa(s, c_0) := \min_{J \subset \{1, \ldots, p\}, |J| \leq s} \min_{M \neq 0} \frac{\|XM\|_F}{\sqrt{n} \|M_J\|_F} > 0.
$$

**Theorem (Oracle inequality)**

Let $\eta > 0$ and $s \leq p$. Consider the estimate $\hat{\Theta} = \hat{P} \hat{T}^t \hat{P}$ with

$$
\lambda = A\sigma \sqrt{\frac{\log p}{n}},
$$

where $A > 4\sqrt{2}$. Then, with probability at least $1 - p^2 - A^2/16$, there exists $C(\eta)$ such that:

$$
\frac{1}{n} \|G\hat{\Theta} - G\Theta^*\|_F^2 \leq

(1 + \eta) \inf_{\Theta, |S_\Theta| \leq s} \left\{ \frac{1}{n} \|G\Theta - G\Theta^*\|_F^2 + \frac{C(\eta)A^2\sigma^2}{\kappa^2(s, 3 + 4/\eta)} \frac{\log p}{n} \right\}
$$
For $P$ fixed

We aim at minimizing the negative penalized log-likelihood:

$$\hat{T} = \arg\min_{T \in \mathcal{T}_P(\mathbb{R})} \left\{ \frac{1}{n} \| G - GPT^t P \|_F^2 + \lambda \| T \|_1 \right\}.$$

- minimization of a convex, differentiable and quadratic function + penalization

$$T_{k+1} = \arg\min_T \left\{ \frac{L}{2} \| T - \left( T_k - \frac{\nabla f(T_k)}{L} \right) \|_F^2 + \lambda \| T \|_1 \right\}.$$

- projection on the space of constraints $\mathcal{T}_P(\mathbb{R})$. 
For $T$ fixed

We aim at minimizing $\hat{P} = \operatorname{argmin}_{P \in \mathcal{P}_p(\mathbb{R})} \left\{ \frac{1}{n} \| G - GPT^t P \|_F^2 \right\}$. Since the space of constraints is not convex, we propose a convex relaxation of the criterion to minimize.

**Definition**

A bistochastic matrix $A = (a_{ij})_{1 \leq i,j \leq p}$ is a matrix such that:

- $a_{ij} \geq 0$,
- $\sum_i a_{ij} = \sum_j a_{ij} = 1$.

**Proposition (Birkhoff)**

The set of bistochastic matrices $\mathcal{B}_p(\mathbb{R})$ is a convex set, which permutation matrices are extreme points.
We can write $\mathcal{B}_p(\mathbb{R}) = \Lambda^+ \cap \mathcal{L}C_1$ as the intersection of the two sets:

1. the convex cone
   \[ \Lambda^+ = \left\{ M = (M^i_j)_{i,j} \in \mathcal{M}_p, \quad \forall i,j, \quad M^i_j \geq 0 \right\}, \]

2. the affine subspace
   \[ \mathcal{L}C_1 = \left\{ M = (M^i_j)_{i,j} \in \mathcal{M}_p, \quad \sum_{i=1}^{p} M^i_j = \sum_{j=1}^{p} M^j_i = 1 \right\}. \]

We use alternate projection algorithms (algorithm of Von Neumann or Boyle-Dykstra) to find the expression of the projected bistochastic matrix.
Algorithm of Boyle-Dykstra
Alternate minimization

\[(\hat{P}, \hat{T}) = \arg\min_{P \in \mathbb{B}_p(\mathbb{R}), T \in \mathbb{T}_p(\mathbb{R})} \frac{1}{n} \| G - GPT^t P \|_F^2 + \lambda \| T \|_1. \]

\[P_0 \in \mathbb{P}_p(\mathbb{R}) \xrightarrow{\text{optimization}} T_0 \xrightarrow{\text{proj}} T_0' \in \mathbb{T}_p(\mathbb{R})\]

projected gradient descent

\[P_1 \in \mathbb{B}_p(\mathbb{R}) \xrightarrow{\text{optimization}} T_1 \xrightarrow{\text{proj}} T_1' \in \mathbb{T}_p(\mathbb{R})\]

projected gradient descent

\[P_2 \in \mathbb{B}_p(\mathbb{R}) \xrightarrow{\text{...}} \text{Projection over } \mathbb{P}_p(\mathbb{R})\]
Permutation matrices

We rewrite the problem of finding the projection of any bistochastic matrix $B \in \mathcal{B}_p(\mathbb{R})$ on $\mathcal{P}_p(\mathbb{R})$ as:

$$\text{Proj}_{\mathcal{P}_p(\mathbb{R})}(B) = \arg\min_{P \in \mathcal{P}_p(\mathbb{R})} \|B - P\|_F$$

$$= \arg\min_{P \in \mathcal{P}_p(\mathbb{R})} -2\langle B, P \rangle_F.$$

Remark that the new function $-2\langle B, P \rangle_F$ to minimize is linear, whereas the space of constraints $\mathcal{P}_p(\mathbb{R})$ is the set of all extreme points of the convex polytope. There exists thus an extreme point solution of the relaxed problem

$$P = \arg\min_{P \in \mathcal{B}_p(\mathbb{R})} -2\langle B, P \rangle_F.$$
Another issue
Joint work with Victor

Instead of relaxing the condition \( P \in \mathbb{P}_p(\mathbb{R}) \), we propose to use genetic algorithms, which are heuristic searches that mimic the process of natural evolution.

\[
\text{Initialization} \rightarrow \text{Selection} \rightarrow \text{Crossover} \rightarrow \text{Mutation} \rightarrow \text{Evaluation} \rightarrow \text{Test}
\]
In few words (initialization)

1. We take $N$ permutation matrices. Each of them will be represented by a sequence of “genes”, called “chromosome”.

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

chromosome $\rightarrow$ \begin{tabular}{cccccc} 2 & 3 & 6 & 1 & 4 & 5 \end{tabular}

2. We search the strict lower-triangular matrix $T$ associated to each chromosome.
In few words (crossover)

1. Selection for the crossover: roulette wheel selection
2. Method of crossover

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
In few words (crossover)

1. Selection for the crossover: roulette wheel selection
2. Method of crossover

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

→

<table>
<thead>
<tr>
<th>4</th>
<th>5</th>
<th>1</th>
<th>3</th>
<th>2</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Performances

Confusion matrix:

<table>
<thead>
<tr>
<th>Reality</th>
<th>Prediction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>edge</td>
<td>no edge</td>
</tr>
<tr>
<td>edge</td>
<td>true positives</td>
<td>false negatives</td>
</tr>
<tr>
<td>no edge</td>
<td>false positives</td>
<td>true negatives</td>
</tr>
</tbody>
</table>

We then define:

- the recall

\[ R = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \]

- the precision

\[ Pr = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \]
Experimental results

We also compute:

- the MSE: $\|\hat{\Theta} - \Theta^*\|_F^2$
- the MSEP: $\frac{1}{n}\| G - G\hat{\Theta}\|_F^2$.

For $n = 100$ and $p = 5$

<table>
<thead>
<tr>
<th></th>
<th>Optimization</th>
<th>G-A</th>
<th>Boosting</th>
<th>Lasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>0.86</td>
<td>0.91</td>
<td>0.91</td>
<td>0.83</td>
</tr>
<tr>
<td>$Pr$</td>
<td>0.63</td>
<td>0.69</td>
<td>0.42</td>
<td>0.46</td>
</tr>
<tr>
<td>MSE</td>
<td>2.62</td>
<td>0.29</td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>MSEP</td>
<td>8.02</td>
<td>4.88</td>
<td>5.26</td>
<td></td>
</tr>
</tbody>
</table>
Experimental results

We also compute:
- the MSE: $\| \hat{\Theta} - \Theta^* \|_F^2$
- the MSEP: $\frac{1}{n} \| G - G\hat{\Theta} \|_F^2$.

For $n = 100$ and $p = 5$

<table>
<thead>
<tr>
<th></th>
<th>Optimization</th>
<th>G-A</th>
<th>Boosting</th>
<th>Lasso</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.86</td>
<td>0.91</td>
<td>0.91</td>
<td>0.83</td>
<td>0.94</td>
</tr>
<tr>
<td>Pr</td>
<td>0.63</td>
<td>0.69</td>
<td>0.42</td>
<td>0.46</td>
<td>0.71</td>
</tr>
<tr>
<td>MSE</td>
<td>2.62</td>
<td>0.29</td>
<td>2.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSEP</td>
<td>8.02</td>
<td>4.88</td>
<td>5.26</td>
<td></td>
<td>4.88</td>
</tr>
</tbody>
</table>