Non-convex inverse problems: project
Sparse PCA

Irène Waldspurger
waldspurger@ceremade.dauphine.fr

Due date: March 21, 2025

General instructions

- You can use any programming language.
- If you reuse code written by someone else, you must cite your source.
- You must provide your code (and it must run without errors!). You should notably include the instructions you used to generate the pictures, so that I can reproduce them.
- If you are stuck, do not hesitate to send me an email describing your problem; I will try to help.
- You can write in either French or English.

Let $k, m, n \in \mathbb{N}^*$ be integers such that $k \leq n$.
Imagine we want to recover a matrix $X \in \mathbb{R}^{n\times n}$. The information we have is threefold:

1. X has rank 1;
2. there exist (unknown) sets $S_1, S_2 \subset \{1, \dots, n\}$ with cardinality k such that, for all $(s_1, s_2) \notin S_1 \times S_2$,
$$X_{s_1,s_2} = 0;$$
3. we have access to $y_i \stackrel{\text{def}}{=} \langle A_i, X \rangle$ for all $i = 1, \dots, m$, where $A_1, \dots, A_m$ are (known) $n \times n$ matrices following independent standard normal distributions.

1. Write a function which selects a random instance of the considered problem. More precisely, given k, m, n, the function
   - selects S_1, S_2 uniformly at random among all subsets of $\{1, \dots, n\}$ with cardinality k;
   - samples random vectors $u, v \in \mathbb{R}^n$ such that $u_s = 0$ for all $s \notin S_1$ and $v_s = 0$ for all $s \notin S_2$;
     [You can choose the random distribution you want for the non-zero entries.]
   - computes $X = uv^T \in \mathbb{R}^n$;
   - samples $A_1, \dots, A_m$ as described above and computes $y_1, \dots, y_m$;
   - returns $X, S_1, S_2, A_1, \dots, A_m, y_1, \dots, y_m$.
2. Write the considered problem under the form of an optimization problem. Is it convex or non-convex? Justify your answer.
3. In this question, we implement a convexified approach. We approximate our problem with a convex optimization problem of the following form:
$$\begin{aligned} &\text{minimize } F(Z), \\ &\text{over all } Z \in \mathbb{R}^{n\times n} \\ &\text{such that } y_k = \langle A_k, Z \rangle, \forall k \leq m, \end{aligned} \qquad \text{(Convex approx.)}$$
   where $F : \mathbb{R}^{n\times n} \to \mathbb{R}$ is a well-chosen objective function.

a) If we had Information 1, but not Information 2, which objective function F from the lectures could you use? We call it F_1.

b) If we had Information 2, but not Information 1, which objective function F from the lectures could you use? We call it F_2.

c) We choose $F = F_1 + \lambda F_2$, for some adequate $\lambda \in \mathbb{R}^+$. Implement a function which, given $\lambda, A_1, \dots, A_m, y_1, \dots, y_m$, solves Problem (Convex approx.).
[You can either write your own solver or use a convex optimization toolbox.]

d) We still denote X the ground true, and S_1, S_2 the sets as in Property 2. We define, for any $Z \in \mathbb{Z}^{n\times n}$, the *support recovery error* of Z as

$$\mathrm{Card}\left\{(s,s') \in \{1,\dots,n\}^2, (s,s') \notin S_1 \times S_2, |Z_{s,s'}| > \frac{\min_X}{10}\right\}$$
$$+\mathrm{Card}\left\{(s,s') \in \{1,\dots,n\}^2, (s,s') \in S_1 \times S_2, |Z_{s,s'}| < \frac{\min_X}{10}\right\},$$

where $\min_X = \min_{(s,s')\in S_1\times S_2} |X_{s,s'}|$.
Write a function which, given X, S_1, S_2, Z, computes the support recovery error of Z.

e) Test your code: take for instance $k = 2, n = 4$. From which value of m do you expect the support recovery error to be always 0, for whatever value of λ? Check that this is indeed how your code behaves. What happens for slightly smaller values of m? [Hint: if the support recovery error is never 0, check your code carefully.]

f) For $k = 3, n = 6, m = 20$, select a random problem instance. Plot the support recovery error as well as $||X - Z||_F$ as a function of λ. Comment the graphs.

g) Set $n = 20$. For $k = 1, 3, 5, ..., 13$, compute approximately the smallest value of m such that, Problem (Convex approx.) reaches zero support recovery error for at least one value of λ with probability roughly 50%. Plot this value as a function of k.

h) Same question if you use only $F = F_1$ or $F = F_2$ as objective function. (Note that, in this case, the objective does not depend on λ.)

i) Comment the plots from the previous two questions.

4. In this question, we implement a non-convex algorithm.

a) Let us set $Z = \frac{1}{m}\sum_{i=1}^m \langle A_i, X\rangle A_i$. Compute $\mathbb{E}(Z)$.

b) Propose a strategy to approximate S_1, S_2 from Z (knowing k), and implement it.

c) Propose a non-convex algorithm which, given a guess for S_1 and S_2, tries to recover X.

If you combine the strategy from Question 4.b) with the algorithm from Question 4.c), you get a complete recovery algorithm.

d) For $k = 4, n = 10$, and various values of m, run the algorithm on several problem instances and print any information you find relevant on either the reconstructed matrix or the inner behavior of the algorithm. Comment the results. Try to explain, in the failure cases, why the algorithm fails.

e) Propose an improvement for this non-convex strategy, describe it and test it.