ESTIMATION OF THE STATE OF A SYSTEM

Andrea Cavallo s320122, Andrea Esposito s319996, Francesco Renis s314803

1 Introduction

The main goal of this activity is to estimate the state of a system in different contexts: in absence or presence of sensors attacks, in target localization problems in a two dimensional indoor area, using sparser observer and in the end in a distributed case.

2 Implementation of IST

First of all, to solve in a proper way the problems described before, we studied the iterative shrinkage/thresholding algorithm (IST). It solves Lasso problems, using a combination of a gradient step and shrinkage/thresholding using the parameter γ as threshold. The algorithm is the following :

$$\mathrm{S}_\alpha(x) = \begin{cases} x-\alpha & if \quad x > \alpha \\ x+\alpha & if \quad x > \alpha \\ 0 & if \quad |x| \leq \alpha \end{cases}$$

And then we applied this algorithm to the function, where we use the gradient

$$x_{t+1} = S_{\lambda\tau}[x_t + \tau A^T(y - Ax_t)]$$

To solve Lasso problem we implemented IST algorithm setting this hyperparameters :

- $\tau = \|C\|_2^{-2} - \epsilon$ with $\epsilon = 10^{-8}$,where C is a q x p matrix generated according to a standard normal distribution : $N(0,1)$;
- Number of sensors q = 10;
- Number of measurements for each sensor p = 20;
- $\Lambda = \lambda * (1.....1)^T$ with $\lambda = \frac{1}{100\tau}$, a vector of dimension p;
- $\gamma = \tau\Lambda = 10^{-1}(1,1,...1)^T$, a vector of dimension p.

Then we generated the support S of the $\tilde{x}$ with uniform distribution and non-zero components i ∈ S such that $\tilde{_i} \in [-2,-1] \cup [1,2]$.

After having determined the measurement noise η with distribution $N(0,\sigma^2)$ and defined the constraint $\|x_{T+1} - x_T\| < \delta$ with $\delta = 10^{-12}$, we ran the algorithm 20 times obtaining the following results:

- The support of $\tilde{x}$ is correctly estimated in the 95% of cases;
- Increasing the number of measurements q, it is not certain that we obtain 100% of success, because we are not sure about the reliability of those measured values due to the presence of possible high noise or errors;
- Setting a $T_{max} = 10^5$ we calculated the number of iterations:

max iterations = 3670;
min iterations = 122;
mean iterations = 710.

- Decreasing τ we can distinguish two cases:
  1. If τ is positive and has a very small value, it has an impact on the estimate calculation because the convergence time becomes larger.
  2. If τ is negative, the conditions of the IST algorithm are inverted, so it is impossible having a sparse vector as result.

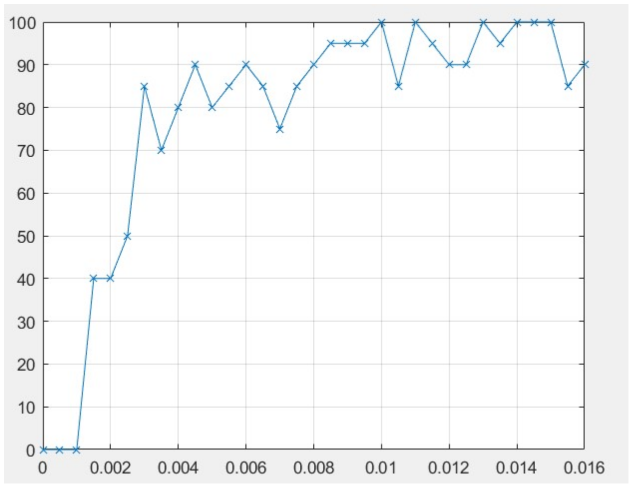

Figure 1: Trend of tau over time

- Decreasing λ the convergence region becomes larger so the greater its value the lower is the accuracy percentage of the estimation.

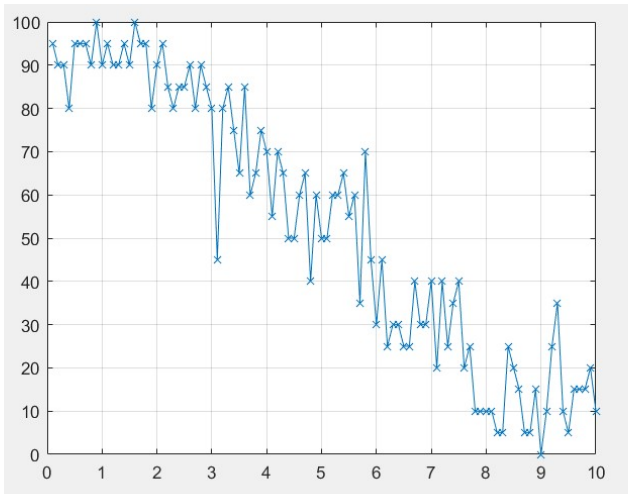

Figure 2: Trend of lambda over time

3 Secure estimation under sparse sensor attacks

We repeated all the passages described before, but adding considering sparse sensor attacks on the value we want to estimate. The given measurement are given as follows:

$$y = C\tilde{x} + \eta + a$$

where a is a vector of q elements. Instead of using x and a as two separate vectors, we unified the in one single vector called z, with dimensions q+p. The attacks can be performed in two ways:

1. Unaware attacks : a_i is uniformly distributed in the intervals $[-2, -1] \cup [1, 2]$
2. Aware attacks: attacker takes the sensor measurements y_i and corrupts it of a quantity equal to $\frac{1}{2}y_i$

We changed some hyperparameters to to be coherent with our z definition:

1. $\tau\Lambda = \tau\lambda(0...0, 1...1)^T \in R^{n+q}$ where n=10 and $\tau\lambda = 2 * 10^{-3}$
2. noise = 0 first and then noise = $N(0, \sigma^2)$

Defining number of sensors n = 10 and number of measurements q = 20, we studied the result obtained using the two types of attacks:

- In the case of Unaware attacks, the algorithm does not converge because the values that affect the measurements can't be neglected, so we are not able to detect the attacks;
- In the case of aware attacks, the estimate values are not properly correct. The results are better than the previous ones because, the values of these attacks are lower, so the impact on the estimate is less significative.

As in the first point, the algorithm is resilient to noises. The main problems are the sparse attacks.

4 Target localization with real data

For this task we used the method of RSS fingerprinting for indoor localization. This method follows two phases:

1. Training phase to create a fingerprint map using a grid of the environment;
2. Runtime phase in which each sensor measures the RSS.

In this particular case we used a provided dictionary D with the RSS fingerprints and a vector y of given measurements, simulating a realistic situation.

Therefore given D $\in R^{q,p}$ and y, we formulated a suitable Lasso for localizing the targets by applying the IST algorithm with the following parameters:

- $\lambda = 1$;
- D normalized, in absence of attacks
- G = (D,I), where I is the identity matrix, normalized in presence of attacks

Adding a single sensor attack on y_i that perturbs the value of $\frac{1}{5}y_i$,we saw that the algorithm is not able to accurately estimate the targets because it detects more than one sensor as under attack, which we know is not the real case.

The efficiency of the algorithm in presence of attacks also depends on which sensor values are corrupted. Since the index of the sensor under attack is chosen randomly at each run, in some cases the corrupted measurements could be closer to other values in the dictionary, increasing the number of fingerprints matched by the algorithm.

5 Sparse observer

In this section we consider a situation with j=4 targets moving in a square room with p=100 cells.

The positions of the targets are stored in a vector x, where $x_i(k) = 1$ if there is a target in the i^{th} cell. In the room there are q = 25 sensors with a random displacement and an RSS fingerprinting method is applied. The model used is the following:

$$x(k+1) = Ax(k)$$
$$y(k) = Dx(k) + \eta + a$$

where A and D are given.
As in the previous point, we used:

- $G = (D, I)$ normalized
- $\hat{z}(k) = [\hat{x}(k), \hat{a}(k)] \in R^{p+q}$
- $\hat{z}(0) = 0$
- η is a measurement noise, $N(0, \sigma^2)$
- $\tau = \|C\|_2^{-2} - \epsilon$ with $\epsilon = 10^{-8}$
- *Lambda* = (10,...,10,20,...,20)
- T at least equal to 50, we used T = 100

The choice of the sensors under attack is random at each run. The attacks can be of two types:

- "Unaware" time-invariant attack
  This type of attack is obtained by setting $a_i = 30$ for each sensor i under attack. In the majority of the runs the algorithm correctly detects the targets and the sensors under attack, mainly because the number of sensors under attack is small relatively to the total number. Moreover we had to tune precisely the time T in order to let the algorithm converge to the final result
- "Aware" time-varying attack
  In this case we consider the sensor measurement y_i to be corrupted of a quantity equal to $\frac{1}{2}y_i$ This kind of attack has a lower impact on the estimation with respect to the unaware case, the algorithm succeeds in detecting the targets in 100% of the cases.

Of course the algorithm is not able to provide a vector containing the estimate of x that is exactly sparse. That is why in order to verify the correctness of the estimation we take the 4 largest values.

Algorithm:
for all k=1,...,T do

$$\hat{z}(k+\tfrac{1}{2}) = S_{\tau\Lambda}[\hat{z}(k) + \tau G^T(y(k) - G\hat{z}(k))]$$
$$\hat{x}(k+1) = A\hat{x}(k+\tfrac{1}{2}$$
$$\hat{a}(k+1) = \hat{a}(k+\tfrac{1}{2})$$

6 Distributed estimation

We consider the situations described in the second section, trying to solve it using a distributed IST algorithm. The goal is to have an estimate $\tilde{x} \in R^n$ in the presence of sparse sensor attacks, given:

$$y = C\tilde{x} + \eta + a$$

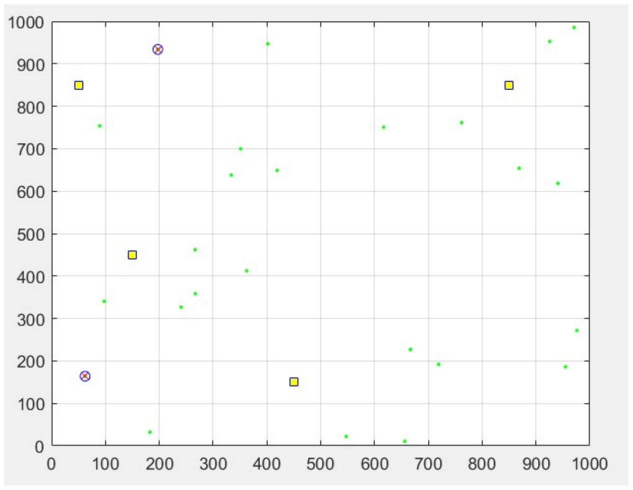

Figure 3: Target (yellow squares) and error (red crosses) localization

We used the following parameters:

- n = 10, number of sensors
- q = 20, number of measurements
- h = 2, sensor attacks
- C $\sim N$(0,1); $\tilde{x} \sim N$(0,1)
- measurement noise $\eta \sim N(0, \sigma^2), \sigma = 10^{-2}$
- Unaware attacks : a_i is uniformly distributed in the intervals $[-2, -1] \cup [1, 2]$
- $\tau\Lambda = \tau\lambda(0, ..., 0, 1..., 1)^T \in R^{n+q}, \tau = 0.03, \lambda = \frac{2*10^{-4}}{\tau}$

With this kind of algorithm, there is no centralized node and each single node computes at each step its own estimation. At the end of the iterations, the estimations of all nodes should as closer as possible. In this case we say that the nodes have achieved consensus.

The algorithm is based on two iterations: for every step of iterations that goes from 1 to T, we implement a cycle that goes from 1 to q in which each node computes its estimate.

```
for all k = 1,...,T do
    for all i = 1,...,q do
```

$$z^{(i)}(k+1) = S_{\tau\Lambda}\left[\sum_{j=1}^{q} Q_{i,j} z^{(i)}(k) + \tau G_i z^{(i)}(k))\right]$$

where Q matrices are given

We repeated the experiment for 20 runs. In the following we present the results divided by the stochastic matrix Q used.

Matrix	Rate of attack detection	Convergence time	Estimation error	Essential spectral radius
Q1	95%	23515	3.5%	0.5857
Q2	95%	32748	3.24%	0.7373
Q3	85%	38542	6.72%	0.9674
Q4	95%	30787	3.05%	0.5103

Table 1: Caption

Analysing the result we obtained that the higher is the essential spectral radius, the more time the algorithm spends to reach the consensus.

The state variable reach the consensus for all the matrices, with a lower error for matrices Q1, Q2 and Q4 and a higher error for Q3 due to its esr, as explained before. Moreover, when we use matrix Q3, the difference between the estimated values of the nodes is in the order of 20%