MODELING AND CONTROL OF CYBER-PHYSICAL SYSTEMS

Note sulla stesura del Report (PART I: MODELING)

Lorenzo AGHILAR (334086), Carlo MIGLIACCIO (332937), Federico PRETINI (329152)

«««< Updated upstream **Deadline** Entro il 28/06/2024
Numero di pagine(max) 4-5 pagine

Task #1: IST Algorithm (Carlo)

Introduzione

- Breve introduzione su ottimizzazione sparsa
- ℓ_0 e sua approssimazione ℓ_1

Algoritmo

- Shrinkage/Thresholding operator
- ISTA e cenni sulla sua derivazione

Risultati

- Grafico Tasso di successo vs numero di sensori q; (potrebbe essere un'idea: slider su LiveScript #sensori)
- Tabella con Valore di τ, Tempi di convergenza, valore di λ, Tasso di successo (evidenziando min, mean, max);
- slider per valori di τ (influisce sul tempo di convergenza) e λ (influisce sulla sparsitÃ della soluzione trovata, supporto);
- Commento su:
  1. Il risultato che ottengo è lo stesso? (NO $\rightarrow$ bias, errore)
  2. Il supporto lo riesco sempre a recuperare?

Task #2: Localization under sparse sensor attacks (Lorenzo)

Introduzione

- LASSO, sparse optimization e CPSs
- Perché uso l'ottimizzazione sparsa per la SSE di CPSs?
- Qualche commento sul setting centralizzato e sull'utilizzo del Fusion Center (ricordare: non ci sono attacchi al fusion center...)

Algoritmo

- Hyperparameters utilizzati
- Estensione del problema del LASSO (pesi λ differenti)
- Algoritmo ISTA per la risoluzione del LASSO

Risultati

- Confronto AWARE vs UNAWARE su:
  - Tasso di rilevamento attacchi
  - Accuratezza della stima

Task #3: Localization under sparse sensor attacks (Lorenzo)

Introduzione

- Indoor localization e RSS fingerprinting
- Perché usiamo la sparsità? Cell-grid discretization...

Algoritmo

- Hyperparameters utilizzati
- Weighted LASSO: λ_1, λ_2 per la soluzione
- K-NN: Svantaggi etc...

Risultati

- Grafico room con sensori e target nei due casi AWARE e UNAWARE (idea: sul Live Script si potrebbe mettere il Menu a tendina per scegliere AWARE/UNAWARE)

Task #4: Dynamic SSE (Federico)

Introduzione

- Breve descrizione del setting dinamico

Algoritmo

- Hyperparameters utilizzati
- Qualche parola su Online Gradient Descent
- Qualche parola su Sparse Observer
- In riferimento agli iperparametri utilizzati mostrare che gli autovalori della matrice $A - L_g C$ siano adeguati (stima asintotica dello stato)

Risultati

- Caso base: Snapshot della stanza in momenti particolari (es: stima completamente errata, stima parzialmente corretta, tracking OK...) (idea: uso del comando `subplot()` su MATLAB)
- (Optional 1) Aware time-variyng attacks: dopo quanto tempo ho convergenza?
- (Optional 2) Sensori sotto attacco che cambiano: dopo quanto tempo riesco ad agganciare di nuovo tutto correttamente?
- (Optional 3) Qual è il limite al numero di sensori? (...qui c'è quel problema da risolvere di stima corretta nonostante ci siano tutti e 25 i sensori sotto attacco)

- Tabella per confrontare Caso baso e Task opzionali 1 e 2 in termini di: (i) converge/non converge, (ii) Tempo di convergenza (dopo quante iterazioni converge)?
- idee per Live Script: (i) Scelta aware/unaware, change sensors con checkbox, (ii) Slider con range (min-max) per sensori sotto attacco, numero di target, T_{max}...

Task #5: Distributed SSE (Carlo)

Introduzione

- Rimozione fusion center
- Vantaggi setting centralizzato e setting distribuito
- Consensus

Algoritmo

- Distributed ISTA: minimizzazione distribuita del funzionale del LASSO (regularization)

Risultati

- Per ogni topologia Q
    1. Autovalori di Q (rispettano il teorema di Perron/Frobenius)
    2. Consensus si/no
    3. Tempo di convergenza e analisi di esr(Q) per ogni tipologia
    4. Tabella con le informazioni precedenti
    5. Grafico che rappresenti la topologia del grafo (Ricorda: prendi Q^T per usare il comando `digraph()`)
- idee per LiveScript: Menu a Tendina per il cambio della topologia...

=======

Task #1: IST Algorithm

Task #2: Secure State Estimation of CPSs

Task #3: Localization under sparse sensor attacks

Task #4: Dynamic SSE

Introduction

In the previous task we performed SSE on a static system, now we leave out the static hypothesis and move on to the dynamic one. Recalling the fact that a CPS with some sensors under attack can be described by means of system (1), in the dynamic case the matrix A is no longer the identity matrix but will become more complex.

Algorithms

In order to solve this problem we could think of using least square or Gradient Descent (**GD**) algorithms but in the former case it might be computationally complex to invert the C-matrix, while in the latter case the algorithm would be too slow to be applied to a dynamic case. In order to speed up the GD we can run a single gradient descent step at each k instant, this algorithm has been called Online Gradient Descent or **OGD**

$$\hat{x}(k+1) = A\hat{x}(k) - \tau AC^T[C\hat{x}(k) - y(k)]$$

Now we can add the attack on the formulation and obtained the so called augmented observability matrix O_t. From the theory we know that if the matrix A as an eigenvalues equal to 1 the dynamic of the CPS also with constant attack is not observable. But thanks to the information about the sparsity of the attack we can develop a SPARSE OBSERVER in order to be able to solve the following problem

min....
after a sufficient small number of step T.
The algorithm of the sparse observer is the following:

Results

Task #5: Distributed SSE

»»»> Stashed changes