Team optimal decentralized filtering with coupled cost

Aditya Mahajan
McGill University

Joint work with Mohammad Afshari

Ninth Workshop on Dynamic Games in Management Science
13 October 2017

Model

Model

One-shot decentralized estimation with coupled cost

Model — State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \text{vec}(\hat{x}_1, \dots, \hat{x}_n)$

Objective — Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

Model

One-shot decentralized estimation with coupled cost

Model — State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \text{vec}(\hat{x}_1, \dots, \hat{x}_n)$

Objective — Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 \\ + q(\hat{x}_1 - \hat{x}_2)^2$$

One-shot decentralized estimation with coupled cost

Model

State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \text{vec}(\hat{x}_1, \dots, \hat{x}_n)$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + q(\hat{x}_1 - \hat{x}_2)^2$$

One-shot decentralized estimation with coupled cost

Model

State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \mathrm{vec}(\hat{x}_1, \dots, \hat{x}_n)$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + (x - \hat{x}_3)^2 + (x - \hat{x}_4)^2$$
$$+ q(\hat{x}_1 - \hat{x}_2)^2 + q(\hat{x}_2 - \hat{x}_3)^2 + q(\hat{x}_3 - \hat{x}_4)^2 + q(\hat{x}_4 - \hat{x}_1)^2$$

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2$$
$$+ q(\hat{x}_1 - \hat{x}_2)^2$$

Model

One-shot decentralized estimation with coupled cost

Model

State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

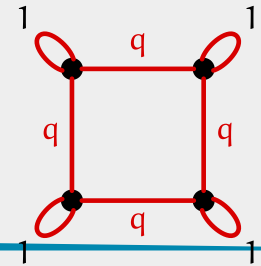
Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \text{vec}(\hat{x}_1, \dots, \hat{x}_n)$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + q(\hat{x}_1 - \hat{x}_2)^2$$

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + (x - \hat{x}_3)^2 + (x - \hat{x}_4)^2 + q(\hat{x}_1 - \hat{x}_2)^2 + q(\hat{x}_2 - \hat{x}_3)^2 + q(\hat{x}_3 - \hat{x}_4)^2 + q(\hat{x}_4 - \hat{x}_1)^2$$



Model

One-shot decentralized estimation with coupled cost

Model

State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \text{vec}(\hat{x}_1, \dots, \hat{x}_n)$

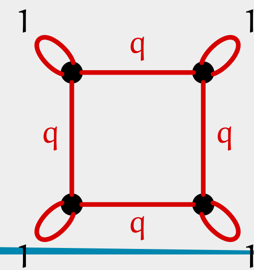
Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$\boxed{c(x, \hat{x}) = \sum_{i \in N} (x - \hat{x}_i)^\top M_{ii} (x - \hat{x}_i) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i - \hat{x}_j)^\top M_{ij} (\hat{x}_i - \hat{x}_j)}$$

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + q(\hat{x}_1 - \hat{x}_2)^2$$

$$+ q(\hat{x}_1 - \hat{x}_2)^2 + q(\hat{x}_2 - \hat{x}_3)^2 + q(\hat{x}_3 - \hat{x}_4)^2 + q(\hat{x}_4 - \hat{x}_1)^2$$



Model

Multi-step decentralized estimation (basic version)

Model

State of the world	:	$x(t+1) = Ax(t) + w^0(t), \quad w^0(t) \sim \mathcal{N}(0, Q_0)$
Observation of agent i	:	$y_i(t) = C_i x(t) + w_i(t), \quad w_i(t) \sim \mathcal{N}(0, Q_i)$
Estimate of agent i	:	$\hat{x}_i(t) = g_{i,t}(y_i(1{:}t))$. Let $\hat{x}(t) = \text{vec}(\hat{x}_1(t), \dots, \hat{x}_n(t))$

Model

Multi-step decentralized estimation (basic version)

Model

State of the world : $x(t+1) = Ax(t) + w^0(t), \quad w^0(t) \sim \mathcal{N}(0, Q_0)$

Observation of agent i: $y_i(t) = C_i x(t) + w_i(t), \quad w_i(t) \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i(t) = g_{i,t}(y_i(1{:}t))$. Let $\hat{x}(t) = \text{vec}(\hat{x}_1(t), \dots, \hat{x}_n(t))$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}\left[\sum_{t=1}^{T} c(x(t), \hat{x}(t))\right]$ where

$$c(x(t), \hat{x}(t)) = \sum_{i \in N} (x(t) - \hat{x}_i(t))^\top M_{ii} (x(t) - \hat{x}_i(t)) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i(t) - \hat{x}_j(t))^\top M_{ij} (\hat{x}_i(t) - \hat{x}_i(t))$$

Model

Multi-step decentralized estimation (basic version)

Model

State of the world : $x(t+1) = Ax(t) + w^0(t), \quad w^0(t) \sim \mathcal{N}(0, Q_0)$

Observation of agent i: $y_i(t) = C_i x(t) + w_i(t), \quad w_i(t) \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i(t) = g_{i,t}(y_i(1{:}t))$. Let $\hat{x}(t) = \text{vec}(\hat{x}_1(t), \dots, \hat{x}_n(t))$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}\left[\sum_{t=1}^{T} c(x(t), \hat{x}(t))\right]$ where

$$c(x(t), \hat{x}(t)) = \sum_{i \in N} (x(t) - \hat{x}_i(t))^\top M_{ii} (x(t) - \hat{x}_i(t)) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i(t) - \hat{x}_j(t))^\top M_{ij} (\hat{x}_i(t) - \hat{x}_i(t))$$

General version

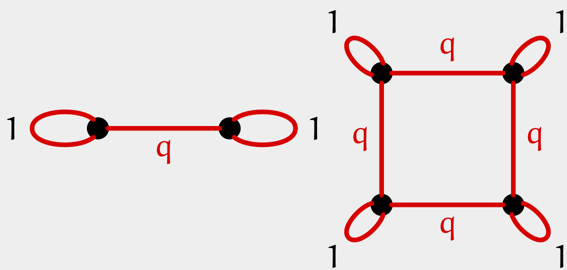
Neighbors can communicate to one another over a communication graph.

$\hat{x}_i(t) = g_i(I_i(t))$, where $I_i(t) = \{y_i(0{:}t), \{I_j(t - d_{ji}\}_{j \in N_i^c}\}$.

Model

Estimation graph

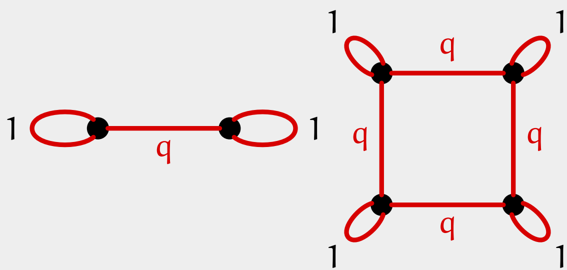
- Weighted undirected graph with self loops
- Weights are positive definite matrices and correspond to the weight given to the error between the estimates of the neighbors.



Model

Estimation graph

- Weighted undirected graph with self loops
- Weights are positive definite matrices and correspond to the weight given to the error between the estimates of the neighbors.



Communication graph

- Weighted directed graph
- Weights are positive integers and correspond to the communication delay between neighbors.

Some representative graphs

- Completely connected graph with d-step delay along each link.
- Strongly connected graph with one-step delay along each link.

Problem Formulation

Given

- The dimension of all system variables
- The covariance of all noise variables
- A communication graph $\mathcal{G}^c$ that determines the information structure
- An estimation graph $\mathcal{G}^e$ that determines the cost coupling between agents

Objective

Choose $g = (g_1, \dots, g_n)$ where $g_i = (g_{i,1}, \dots, g_{i,T})$ and

$$\boxed{\hat{x}_i(t) = g_{i,t}(I_i(t))}$$

to minimize $\mathbb{E}\left[\sum_{t=1}^{T} c(x(t), \hat{x}(t))\right]$ where

$$\boxed{c(x(t), \hat{x}(t)) = \sum_{i \in N} (x(t) - \hat{x}_i(t))^\top M_{ii} (x(t) - \hat{x}_i(t)) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i(t) - \hat{x}_j(t))^\top M_{ij} (\hat{x}_i(t) - \hat{x}_i(t))}$$

Motivation

Decentralized Control

Does separation of estimation and control hold for decentralized control systems?

Motivation

Decentralized Control

Does separation of estimation and control hold for decentralized control systems?

Consensus in sensor networks

Lot of recent literature on ad-hoc iterative algorithms in which agents converge to a consensus solution. In the proposed model, the cost is chosen such that consensus will emerge naturally.

Motivation

Decentralized Control	Does separation of estimation and control hold for decentralized control systems?
Consensus in sensor networks	Lot of recent literature on ad-hoc iterative algorithms in which agents converge to a consensus solution. In the proposed model, the cost is chosen such that consensus will emerge naturally.
Estimation in vehicle patoons	Each vehicle needs to estimate the location of all vehicles in a platoon.
Other potential applications	. . .

Key Observation

The decentralized filtering problem is a static team

An overview of static teams

An overview of static teams

Brief literature overview

Research on **static teams** started in Economics in the context of organizational behavior

- Marschack(1950's), Radnar (1962), Marschack and Radnar (1972)

Dynamic teams have been studied in Systems and Control since late 60's

- Witensenhausen (1969): A two-step LQG system with two controllers. **Non-linear controllers outperform linear control strategies**. Finding the optimal controller for this model is still an open problem.
- Whittle and Rudge (1974): Infinite horizon LQG model with two symmetric controllers. A priori restrict attention to linear controllers. **Best linear controllers cannot be represented by recursions of finite order**.
- Some positive results: Witsenhausen (1971), Ho and Chu (1972), Krainak Speyer Marcus (1982), Aicardi Davoli Minciardi (1987), Nayyar Mahajan Teneketzis (2013).

A simplified version of Radner's model

Model

- Decentralized system with n agents.
- $(x, y_1, \dots, y_n)$ jointly Gaussian. $\operatorname{cov}(x, y_i) = \Theta_i$, $\operatorname{cov}(y_i, y_j) = \Sigma_{ij}$.
- Agent i observes y_i and chooses $u_i = g_i(y_i)$.

- Radner, "Team Decision Problems," Annals of Math. Stats, 1962.

An overview of static teams

A simplified version of Radner's model

Model

- Decentralized system with n agents.
- $(x, y_1, \dots, y_n)$ jointly Gaussian. $\text{cov}(x, y_i) = \Theta_i$, $\text{cov}(y_i, y_j) = \Sigma_{ij}$.
- Agent i observes y_i and chooses $u_i = g_i(y_i)$.

Objective

Choose $g = (g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, u)]$ where

$$c(x, u) = \mathbb{E}\Big[\sum_{i,j \in N} (u_i)^\top R_{ij} u_j + 2 \sum_{i \in N} (u_i)^\top P_i x \Big]$$

- Radner, "Team Decision Problems," Annals of Math. Stats, 1962.

The idea of Radner's solution

Necessary condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal only if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}_i, g_{-i}) - J(g) \geqslant 0$$

This also implies that the strategy g is person by person optimal.

Sufficient condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}) - J(g) \geqslant 0$$

The idea of Radner's solution

Necessary condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal only if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}_i, g_{-i}) - J(g) \geqslant 0$$

This also implies that the strategy g is person by person optimal.

Sufficient condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}) - J(g) \geqslant 0$$

Radner's key result was to show that PBPO implies team optimality.

The idea of Radner's solution

Necessary condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal only if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}_i, g_{-i}) - J(g) \geqslant 0$$

This also implies that the strategy g is person by person optimal.

Sufficient condition for optimality

A strategy $g = (g_1, \dots, g_n)$ is optimal if for any other strategy $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_n)$

$$J(\tilde{g}) - J(g) \geqslant 0$$

> Radner's key result was to show that PBPO implies team optimality.

Necessary and sufficient condition

$$g_i(y_i) = u_i \text{ such that } \frac{\partial}{\partial u_i} \mathbb{E}[c(x, g_{-i}(y_{-i}), u_i)) \,|\, y_i] = 0$$

Radner's solution (cont.)

Main result

Optimal control law is linear and is given by

$$\boxed{u^i = F^i(y^i - \mathbb{E}[y^i]) + H^i\,\mathbb{E}[x],}$$

where

$$\boxed{F = -\Gamma^{-1}\eta, \qquad H = -R^{-1}P,}$$

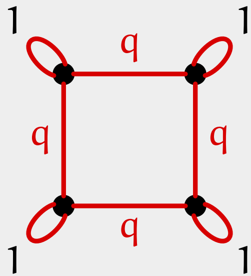
and

- $F = \text{vec}(F^1, F^2, \cdots, F^n)$
- $H = \text{rows}(H^1, H^2, \cdots, H^n)$.
- $\Gamma = [\Gamma^{ij}]$, where $\Gamma^{ij} = \Sigma^{ij} \otimes R^{ij}$.
- $\eta = \text{vec}(P^1\Theta^1, P^2\Theta^2, \dots, P^n\Theta^n)$.

One-step decentralized estimation

One-step decentralized estimation

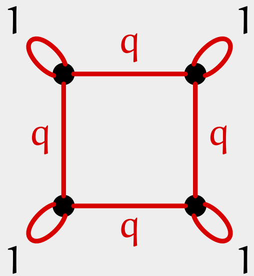
One-step decentralized estimation as a static team



In the decentralized estimation problem, we have

$$c(x, \hat{x}) = \sum_{i \in N} (x - \hat{x}_i)^\top M_{ii} (x - \hat{x}_i) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i - \hat{x}_j)^\top M_{ij} (\hat{x}_i - \hat{x}_j)$$

One-step decentralized estimation as a static team



In the decentralized estimation problem, we have

$$c(x, \hat{x}) = \sum_{i \in N} (x - \hat{x}_i)^\top M_{ii} (x - \hat{x}_i) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i - \hat{x}_j)^\top M_{ij} (\hat{x}_i - \hat{x}_j)$$

This can be written as $\boxed{x^\top Q x + \hat{x}^\top R \hat{x} + 2\hat{x}^\top P x}$, where

- $Q = \sum_{i \in N} M_{ii}$,
- $P = \text{rows}(-M_{ii}, \dots, -M_{nn})$
- $R = [R_{ij}]$, where

$$R_{ij} = \begin{cases} M_{ii} + \sum_{j \in N_j} M_{ij}, & \text{if } i = j \\ -M_{ij}, & \text{if } j \in N_i \\ 0, & \text{otherwise} \end{cases} \qquad \text{(Graph Laplacian)}$$

- $\Sigma_{ii} = C_i \Sigma_x (C_i)^\top + \text{var}(w_i)$
- $\Sigma_{ij} = C_i \Sigma_x (C_j)^\top$
- $\Theta_i = \Sigma_x (C_i)^\top$.

One-step decentralized estimation

Optimal solution for one-shot decentralized estimation

Translating Radner's result

Since the model is a static team, from Radner's result we can say that the optimal estimates are

$$\hat{x}_i = F_i y_i$$

However, this form of the solution does not work well for the multi-step case.

One-step decentralized estimation

Optimal solution for one-shot decentralized estimation

Translating Radner's result

Since the model is a static team, from Radner's result we can say that the optimal estimates are

$$\hat{x}_i = F_i y_i$$

However, this form of the solution does not work well for the multi-step case.

An alternative form of the solution

Let $\hat{x}_i^{\text{loc}} = \mathbb{E}[x \mid y_i]$. Then, the optimal estimates are given by

$$\boxed{\hat{x}_i = L_i \, \hat{x}_i^{\text{loc}}, \quad L = -\Gamma^{-1}\eta}$$

- $L = \text{vec}(L^1, \dots, L^n)$
- $\hat{\Sigma}_{ij} = \text{cov}(\hat{x}_i, \hat{x}_j) = \Theta_i(\Sigma_{ii})^{-1}\Sigma_{ij}(\Sigma_{jj})^{-1}(\Theta_j)^\top$
- $\Gamma = [\Gamma_{ij}]$, where $\Gamma_{ij} = \hat{\Sigma}_{ij} \otimes R_{ij}$
- $\eta = \text{vec}(P_1\hat{\Sigma}_{11}, \dots, P_n\hat{\Sigma}_{nn})$

One-step decentralized estimation

One-step decentralized estimation as a static team

Optimal solution for one-shot decentralized estimation

Examples of one-shot estimation

Suppose $x \sim \mathcal{N}(0, \sigma_0^2)$ and $y_i = x + w_i$ where $w_i \sim \mathcal{N}(0, \sigma^2)$. Then,

$$\Gamma = \begin{bmatrix} 1+q & -\alpha q \\ -\alpha q & 1+q \end{bmatrix}, \quad \text{where } \alpha = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}.$$

One-step decentralized estimation

Examples of one-shot estimation

Suppose $x \sim \mathcal{N}(0, \sigma_0^2)$ and $y_i = x + w_i$ where $w_i \sim \mathcal{N}(0, \sigma^2)$. Then,

$$\Gamma = \begin{bmatrix} 1+q & -\alpha q \\ -\alpha q & 1+q \end{bmatrix}, \quad \text{where } \alpha = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}.$$

$$\Gamma^{-1} = \frac{1}{(1+q)^2 - (\alpha q)^2} \begin{bmatrix} 1+q & \alpha q \\ \alpha q & 1+q \end{bmatrix}.$$

One-step decentralized estimation

Examples of one-shot estimation

$$\hat{x}_i = \frac{1}{1+\bar{\alpha} q}\,\hat{x}_i^{\mathrm{loc}}, \quad \text{where } \bar{\alpha} = \frac{\sigma^2}{\sigma_0^2+\sigma^2}.$$

One-step decentralized estimation

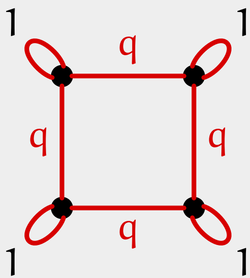
One-step decentralized estimation as a static team

Optimal solution for one-shot decentralized estimation

Examples of one-shot estimation

Examples of one-shot estimation

$$\hat{x}_i = \frac{1}{1 + \bar{\alpha} q} \hat{x}_i^{\mathrm{loc}}, \quad \text{where } \bar{\alpha} = \frac{\sigma^2}{\sigma_0^2 + \sigma^2}.$$

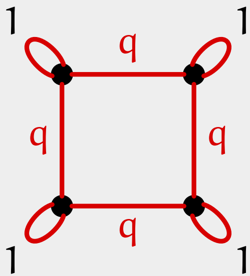


$$\hat{x}_i = \frac{1}{1 + 2\bar{\alpha} q} \hat{x}_i^{\mathrm{loc}}.$$

One-step decentralized estimation

Examples of one-shot estimation

$$\hat{x}_i = \frac{1}{1 + \bar{\alpha} q} \hat{x}_i^{\text{loc}}, \quad \text{where } \bar{\alpha} = \frac{\sigma^2}{\sigma_0^2 + \sigma^2}.$$



$$\hat{x}_i = \frac{1}{1 + 2\bar{\alpha} q} \hat{x}_i^{\text{loc}}.$$

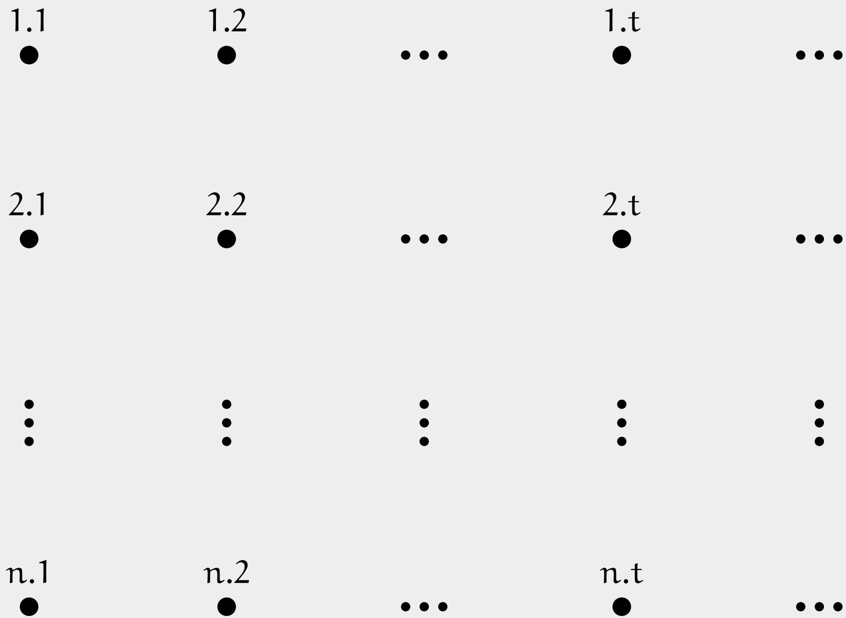
d-regular graph

$$\hat{x}_i = \frac{1}{1 + d\bar{\alpha} q} \hat{x}_i^{\text{loc}}.$$

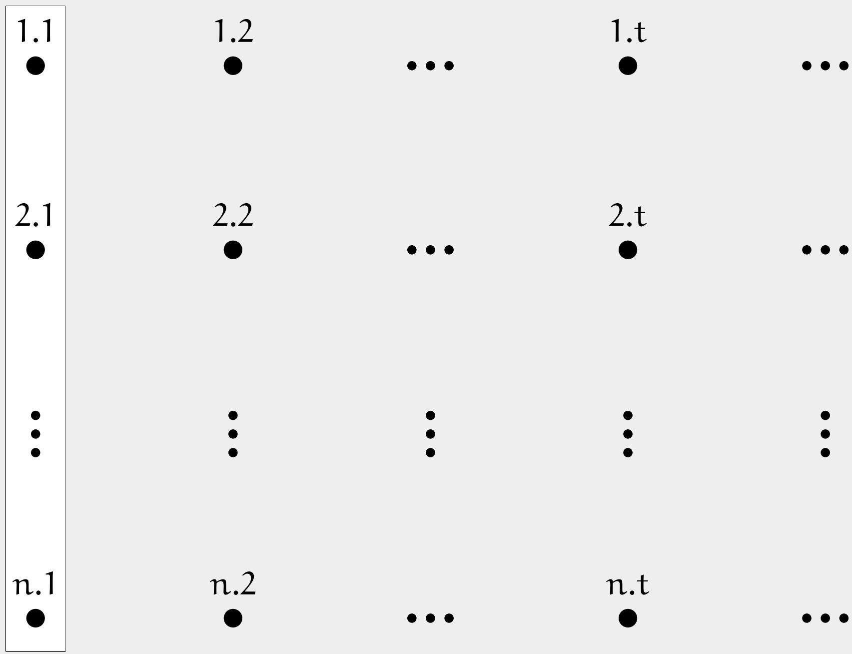
Proof: Show that $\Gamma L = -\eta$

Multi-step decentralized estimation

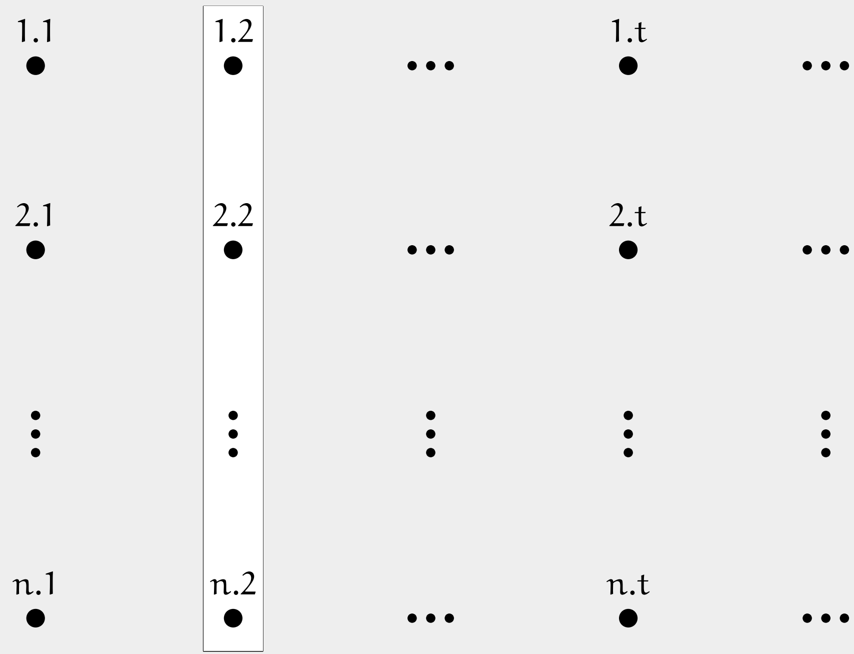
Multi-step decentralized estimation



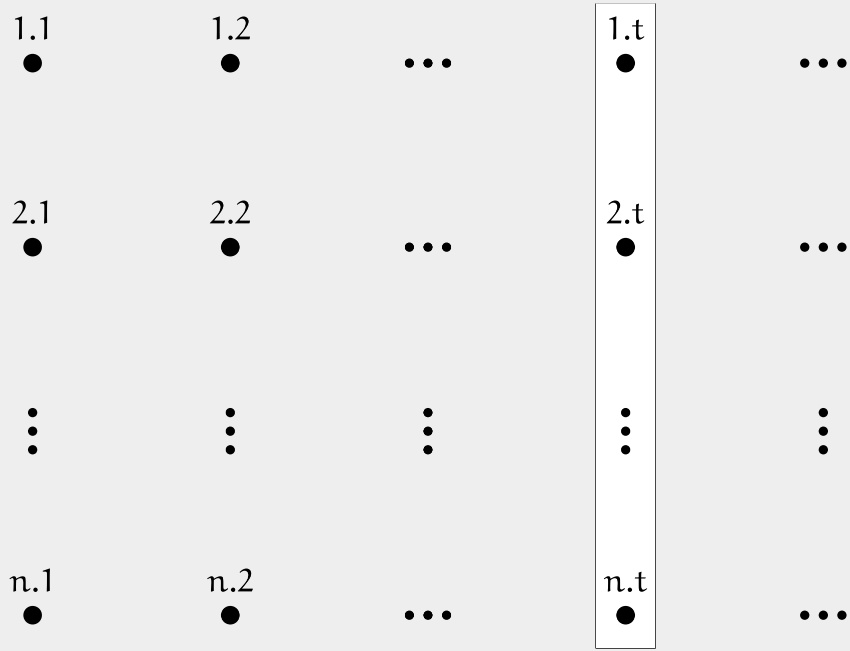
Multi-step decentralized estimation



Multi-step decentralized estimation



Multi-step decentralized estimation



Multi-step decentralized estimation

1.1 ● 1.2 ● … $1.t$ ● …

Instead of solving $\min \mathbb{E}\left[\sum_{t=1}^{T} c(x(t), \hat{x}(t))\right]$

we can solve $\min \mathbb{E}[c(x(t), \hat{x}(t))]$ for each t.

$n.1$ ● $n.2$ ● … $n.t$ ● …

Multi-step decentralized estimation

Key observation The problem at time t is a one-shot optimization problem

Multi-step decentralized estimation

Key observation: The problem at time t is a one-shot optimization problem

Optimal estimator: Let $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid I_i(t)]$ and $\hat{\Sigma}_{ij}(t) = \text{cov}(\hat{x}_i^{\text{loc}}(t), \hat{x}_j^{\text{loc}}(t))$. Then,

$$\hat{x}_i(t) = L_i(t)\, \hat{x}_i^{\text{loc}}(t),$$

$$\text{vec}(L_i(t)) = -\left[\hat{\Sigma}_{ij}(t) \otimes R_{ij}\right]^{-1} \text{vec}(P_i \hat{\Sigma}_{ii}(t))$$

Multi-step decentralized estimation

Key observation The problem at time t is a one-shot optimization problem

Optimal estimator Let $\hat{x}^{\text{loc}}_i(t) = \mathbb{E}[x(t) \,|\, I_i(t)]$ and $\hat{\Sigma}_{ij}(t) = \text{cov}(\hat{x}^{\text{loc}}_i(t), \hat{x}^{\text{loc}}_j(t))$. Then,

$$\hat{x}_i(t) = L_i(t)\, \hat{x}^{\text{loc}}_i(t),$$
$$\text{vec}(L_i(t)) = -\big[\hat{\Sigma}_{ij}(t) \otimes R_{ij}\big]^{-1} \text{vec}(P_i \hat{\Sigma}_{ii}(t))$$

Remarks To compute the optimal solution, we only need to compute $\hat{x}^{\text{loc}}_i(t)$ and $\hat{\Sigma}_{ij}(t)$.

Recall, all random variables are jointly Gaussian. Pre-computing $\hat{\Sigma}_{ij}(t)$ and keeping track of $\hat{x}^{\text{loc}}_i(t)$ is trivial but for computational complexity.

Almost same as standard Kalman filtering! Relatively straight forward to come up with recursive equations (but for notation!).

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ij}(t)$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples . . .

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples . . .

One step delay sharing

Complete communication graph with one unit communication delay.

$$I_i(t) = \{y_i(t), y(1{:}t-1)\}$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples . . .

One step delay sharing

Complete communication graph with one unit communication delay.

$$I_i(t) = \{y_i(t), y(1{:}t-1)\}$$

d-step delay sharing

Complete communication graph with d units communication delay.

$$I_i(t) = \{y_i(t-d+1{:}t), y(1{:}t-d)\}$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples . . .

One step delay sharing

Complete communication graph with one unit communication delay.

$$I_i(t) = \{y_i(t), y(1{:}t-1)\}$$

d-step delay sharing

Complete communication graph with d units communication delay.

$$I_i(t) = \{y_i(t-d+1{:}t), y(1{:}t-d)\}$$

General comm. graph

Assume a strongly connected (directed) communication graph.

Can be effectively viewed as a d-step delay sharing, where d is the diameter of the graph.

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

One-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \,|\, y_i(t), y(1{:}t-1)]$. Define $\hat{x}^{\text{com}}(t) = \mathbb{E}[x(t) \,|\, y(1{:}t-1)]$. Then,

$$\hat{x}_i^{\text{loc}}(t) = \hat{x}^{\text{com}}(t) + K_i(t)\big[y_i(t) - C_i(t)\hat{x}^{\text{com}}(t)\big]$$

One-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid y_i(t), y(1{:}t-1)]$. Define $\hat{x}^{\text{com}}(t) = \mathbb{E}[x(t) \mid y(1{:}t-1)]$. Then,

$$\hat{x}_i^{\text{loc}}(t) = \hat{x}^{\text{com}}(t) + K_i(t)\big[y_i(t) - C_i(t)\hat{x}^{\text{com}}(t)\big]$$

$$\hat{x}^{\text{com}}(t+1) = A\hat{x}^{\text{com}}(t) + AK(t)\big[y(t) - C\hat{x}^{\text{com}}(t)\big]$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

One-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \,|\, y_i(t), y(1{:}t-1)]$. Define $\hat{x}^{\text{com}}(t) = \mathbb{E}[x(t) \,|\, y(1{:}t-1)]$. Then,

$$\hat{x}_i^{\text{loc}}(t) = \hat{x}^{\text{com}}(t) + K_i(t)\big[y_i(t) - C_i(t)\hat{x}^{\text{com}}(t)\big]$$

$$\hat{x}^{\text{com}}(t+1) = A\hat{x}^{\text{com}}(t) + AK(t)\big[y(t) - C\hat{x}^{\text{com}}(t)\big]$$

Recursion for conditional covariance

Let $\Sigma(t) = \text{var}(x(t) - \hat{x}^{\text{com}}(t))$. The gains are given by

$$K_i(t) = \Sigma(t)(C_i(t))^\top\big[C_i\Sigma(t)(C_i)^\top + \text{var}(w_i)\big]^{-1}$$

$$K(t) = \Sigma(t)C^\top\big[C\Sigma(t)C^\top + \text{var}(w_1, \dots, w_n)\big]$$

Define $\Lambda(t) = I - K(t)C$.

$$\Sigma(t) = A\Lambda(t)\Sigma(t)\Lambda(t)^\top A^\top + \text{var}(w_0) + AK(t)\,\text{var}(w_1, \dots, w_n)K(t)^\top A^\top$$

One-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid y_i(t), y(1{:}t-1)]$. Define $\hat{x}^{\text{com}}(t) = \mathbb{E}[x(t) \mid y(1{:}t-1)]$. Then,

$$\hat{x}_i^{\text{loc}}(t) = \hat{x}^{\text{com}}(t) + K_i(t)\big[y_i(t) - C_i(t)\hat{x}^{\text{com}}(t)\big]$$

$$\hat{x}^{\text{com}}(t+1) = A\hat{x}^{\text{com}}(t) + AK(t)\big[y(t) - C\hat{x}^{\text{com}}(t)\big]$$

Recursion for conditional covariance

Let $\Sigma(t) = \text{var}(x(t) - \hat{x}^{\text{com}}(t))$. The gains are given by

$$K_i(t) = \Sigma(t)(C_i(t))^\top \big[C_i\Sigma(t)(C_i)^\top + \text{var}(w_i)\big]^{-1}$$

$$K(t) = \Sigma(t)C^\top \big[C\Sigma(t)C^\top + \text{var}(w_1, \dots, w_n)\big]$$

Standard Kalman filter

Define $\Lambda(t) = I - K(t)C$.

$$\Sigma(t) = A\Lambda(t)\Sigma(t)\Lambda(t)^\top A^\top + \text{var}(w_0) + AK(t)\,\text{var}(w_1, \dots, w_n)K(t)^\top A^\top$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ij}(t)$

Proof by examples

One-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \,|\, y_i(t), y(1{:}t-1)]$. Define $\hat{x}^{\text{com}}(t) = \mathbb{E}[x(t) \,|\, y(1{:}t-1)]$. Then,

$$\hat{x}_i^{\text{loc}}(t) = \hat{x}^{\text{com}}(t) + K_i(t)\big[y_i(t) - C_i(t)\hat{x}^{\text{com}}(t)\big]$$

$$\hat{x}^{\text{com}}(t+1) = A\hat{x}^{\text{com}}(t) + AK(t)\big[y(t) - C\hat{x}^{\text{com}}(t)\big]$$

Recursion for conditional covariance

Let $\Sigma(t) = \text{var}(x(t) - \hat{x}^{\text{com}}(t))$. The gains are given by

$$K_i(t) = \Sigma(t)(C_i(t))^\top\big[C_i\Sigma(t)(C_i)^\top + \text{var}(w_i)\big]^{-1}$$

$$K(t) = \Sigma(t)C^\top\big[C\Sigma(t)C^\top + \text{var}(w_1, \dots, w_n)\big]$$

Standard Kalman filter

Define $\Lambda(t) = I - K(t)C$.

$$\Sigma(t) = A\Lambda(t)\Sigma(t)\Lambda(t)^\top A^\top + \text{var}(w_0) + AK(t)\,\text{var}(w_1, \dots, w_n)K(t)^\top A^\top$$

Covariance across agents

$$\hat{\Sigma}_{ij}(t) = K_i C_i \Sigma(t)(C_j)^\top (K_j)^\top$$

d-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid y_i(t-d+1{:}t), y(1{:}t-d)]$.
Define $\hat{x}^{\text{com}}(t-d+1) = \mathbb{E}[x(t-d+1) \mid y(1{:}t-d)]$.

$$\hat{x}_i^{\text{loc}}(t) = A^{d-1}\hat{x}^{\text{com}}(t-d+1) + K_i(t)\left\{\begin{bmatrix} y_i(t) \\ y_i(t-1) \\ \vdots \\ y_i(t-d+1) \end{bmatrix} - \underbrace{\begin{bmatrix} C_i(t)A^{d-1} \\ C_i(t)A^{d-2} \\ \vdots \\ C_i(t) \end{bmatrix}}_{\bar{C}_i(t)} \hat{x}^{\text{com}}(t-d+1)\right\}$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples

One-step delay sharing

d-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid y_i(t-d+1{:}t), y(1{:}t-d)]$.

Define $\hat{x}^{\text{com}}(t-d+1) = \mathbb{E}[x(t-d+1) \mid y(1{:}t-d)]$.

$$\hat{x}_i^{\text{loc}}(t) = A^{d-1}\hat{x}^{\text{com}}(t-d+1) + K_i(t)\left\{ \begin{bmatrix} y_i(t) \\ y_i(t-1) \\ \vdots \\ y_i(t-d+1) \end{bmatrix} - \underbrace{\begin{bmatrix} C_i(t)A^{d-1} \\ C_i(t)A^{d-2} \\ \vdots \\ C_i(t) \end{bmatrix}}_{\bar{C}_i(t)} \hat{x}^{\text{com}}(t-d+1) \right\}$$

Standard Kalman filter

$$\hat{x}^{\text{com}}(t+1) = A\hat{x}^{\text{com}}(t) + AK_t\left[y_t - C\hat{x}^{\text{com}}(t)\right]$$

and

$$\Sigma(t) = A\Lambda(t)\Sigma(t-1)\Lambda(t)^\top A^\top + \text{var}(w_0) + AK(t)\,\text{var}(w_1, \dots, w_n)K(t)^\top A^\top$$

Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples

One-step delay sharing

d-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid y_i(t-d+1{:}t), y(1{:}t-d)]$.

Define $\hat{x}^{\text{com}}(t-d+1) = \mathbb{E}[x(t-d+1) \mid y(1{:}t-d)]$.

$$\hat{x}_i^{\text{loc}}(t) = A^{d-1}\hat{x}^{\text{com}}(t-d+1) + K_i(t)\left\{ \begin{bmatrix} y_i(t) \\ y_i(t-1) \\ \vdots \\ y_i(t-d+1) \end{bmatrix} - \underbrace{\begin{bmatrix} C_i(t)A^{d-1} \\ C_i(t)A^{d-2} \\ \vdots \\ C_i(t) \end{bmatrix}}_{\bar{C}_i(t)} \hat{x}^{\text{com}}(t-d+1) \right\}$$

Recursion for conditional covariance

$$K_i(t) = \left[A^{d-1}\Sigma(t-d)(\bar{C}_i)^{\intercal} + \bar{\Sigma}_{i0}(t-k+1)\right]\left[\bar{C}_i\Sigma(t-d)(\bar{C}_i)^{\intercal} + \bar{\Sigma}_{ii}(t-d+1)\right]^{-1}$$

where $\bar{w}_i(t-d+1) = W_i \operatorname{vec}(w_i(t), \dots, w_i(t-d+1))$

and $\bar{\Sigma}_{ij}(t) = \operatorname{cov}(\bar{w}_i(t), \bar{w}_j(t))$.

d-step delay sharing

Recursion for local estimates

Recall $\hat{x}_i^{loc}(t) = \mathbb{E}[x(t) \mid y_i(t-d+1{:}t), y(1{:}t-d)]$.

Define $\hat{x}^{com}(t-d+1) = \mathbb{E}[x(t-d+1) \mid y(1{:}t-d)]$.

$$\hat{x}_i^{loc}(t) = A^{d-1}\hat{x}^{com}(t-d+1) + K_i(t)\left\{ \begin{bmatrix} y_i(t) \\ y_i(t-1) \\ \vdots \\ y_i(t-d+1) \end{bmatrix} - \underbrace{\begin{bmatrix} C_i(t)A^{d-1} \\ C_i(t)A^{d-2} \\ \vdots \\ C_i(t) \end{bmatrix}}_{\bar{C}_i(t)} \hat{x}^{com}(t-d+1) \right\}$$

Recursion for conditional covariance

$$K_i(t) = \left[A^{d-1}\Sigma(t-d)(\bar{C}_i)^\top + \bar{\Sigma}_{i0}(t-k+1)\right]\left[\bar{C}_i\Sigma(t-d)(\bar{C}_i)^\top + \bar{\Sigma}_{ii}(t-d+1)\right]^{-1}$$

where $\bar{w}_i(t-d+1) = W_i \operatorname{vec}(w_i(t), \dots, w_i(t-d+1))$
and $\bar{\Sigma}_{ij}(t) = \operatorname{cov}(\bar{w}_i(t), \bar{w}_j(t))$.

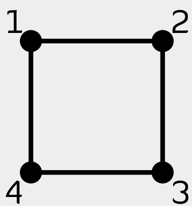
Covariance across agents

$$\hat{\Sigma}_{ij}(t) = K_i(t)\left[\bar{C}_i\Sigma(t-d)(\bar{C}_j)^\top + \operatorname{cov}(\bar{w}_i, \bar{w}_j)\right](K_j(t))^\top$$

Recursive computation of $\hat{x}_i^{\mathrm{loc}}(t)$ and $\hat{\Sigma}_{ij}(t)$

General graph

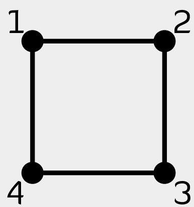
Information structure $\quad I_1(t) = \{y_1(1:t), y_2(1{:}t-1), y_3(1{:}t-2), y_4(1{:}t-1)\}$



General graph

Information structure

$$I_1(t) = \{y_1(1:t), y_2(1{:}t-1), y_3(1{:}t-2), y_4(1{:}t-1)\}$$
$$= \{\underbrace{y_1(t), y_1(t-1), y_2(t-1), y_4(t-1)}_{\text{local info}}, \underbrace{y(1{:}t-2)}_{\text{common info}}\}$$



Recursive computation of $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ii}(t)$

Proof by examples

One-step delay sharing

d-step delay sharing

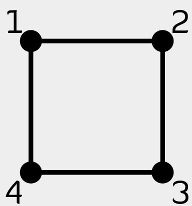
General graph

Information structure

$$I_1(t) = \{y_1(1:t), y_2(1:t-1), y_3(1:t-2), y_4(1:t-1)\}$$
$$= \{\underbrace{y_1(t), y_1(t-1), y_2(t-1), y_4(t-1)}_{\text{local info}}, \underbrace{y(1:t-2)}_{\text{common info}}\}$$

Local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid I_i(t)]$. Then,



$$\hat{x}_1^{\text{loc}}(t) = A\hat{x}^{\text{com}}(t-1) + K_1(t)\left\{\begin{bmatrix} y_1(t) \\ y_1(t-1) \\ y_2(t-1) \\ y_4(t-1) \end{bmatrix} - \begin{bmatrix} C_1 A \\ C_1 \\ C_2 \\ C_4 \end{bmatrix}\hat{x}^{\text{com}}(t-1)\right\}$$

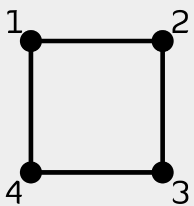
General graph

Information structure

$$I_1(t) = \{y_1(1:t), y_2(1{:}t-1), y_3(1{:}t-2), y_4(1{:}t-1)\}$$
$$= \{\underbrace{y_1(t), y_1(t-1), y_2(t-1), y_4(t-1)}_{\text{local info}}, \underbrace{y(1{:}t-2)}_{\text{common info}}\}$$

Local estimates

Recall $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid I_i(t)]$. Then,



$$\hat{x}_1^{\text{loc}}(t) = A\hat{x}^{\text{com}}(t-1) + K_1(t)\left\{\begin{bmatrix} y_1(t) \\ y_1(t-1) \\ y_2(t-1) \\ y_4(t-1) \end{bmatrix} - \begin{bmatrix} C_1 A \\ C_1 \\ C_2 \\ C_4 \end{bmatrix}\hat{x}^{\text{com}}(t-1)\right\}$$

Remarks

- Effectively equivalent to d-step delayed sharing.
- Each node keeps track of a delayed centralized estimator and innovation wrt common information.

Summary

Summary

One-shot decentralized estimation with coupled cost

Model

State of the world : $x \sim \mathcal{N}(0, \Sigma_x)$

Observation of agent i: $y_i = C_i x + w_i, \quad w_i \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i = g_i(y_i)$. Let $\hat{x} = \mathrm{vec}(\hat{x}_1, \dots, \hat{x}_n)$

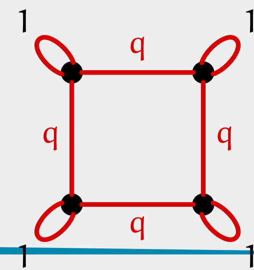
Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}[c(x, \hat{x})]$ where . . .

$$c(x, \hat{x}) = \sum_{i \in N} (x - \hat{x}_i)^\top M_{ii} (x - \hat{x}_i) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i - \hat{x}_j)^\top M_{ij} (\hat{x}_i - \hat{x}_j)$$

$$(x - \hat{x}_1)^2 + (x - \hat{x}_2)^2 + q(\hat{x}_1 - \hat{x}_2)^2$$

$$+ q(\hat{x}_1 - \hat{x}_2)^2 + q(\hat{x}_2 - \hat{x}_3)^2 + q(\hat{x}_3 - \hat{x}_4)^2 + q(\hat{x}_4 - \hat{x}_1)^2$$



Summary

One-shot decentralized estimation with coupled cost

Multi-step decentralized estimation (basic version)

Model

State of the world : $x(t+1) = Ax(t) + w^0(t), \quad w^0(t) \sim \mathcal{N}(0, Q_0)$

Observation of agent i: $y_i(t) = C_i x(t) + w_i(t), \quad w_i(t) \sim \mathcal{N}(0, Q_i)$

Estimate of agent i : $\hat{x}_i(t) = g_{i,t}(y_i(1{:}t))$. Let $\hat{x}(t) = \text{vec}(\hat{x}_1(t), \dots, \hat{x}_n(t))$

Objective

Choose $(g_1, \dots, g_n)$ to minimize $\mathbb{E}\left[\sum_{t=1}^{T} c(x(t), \hat{x}(t))\right]$ where

$$\boxed{c(x(t), \hat{x}(t)) = \sum_{i \in N} (x(t) - \hat{x}_i(t))^\top M_{ii} (x(t) - \hat{x}_i(t)) + \frac{1}{2} \sum_{i,j \in N} (\hat{x}_i(t) - \hat{x}_j(t))^\top M_{ij} (\hat{x}_i(t) - \hat{x}_i(t))}$$

General version

Neighbors can communicate to one another over a communication graph.

$\hat{x}_i(t) = g_i(I_i(t))$, where $I_i(t) = \{y_i(0{:}t), \{I_j(t - d_{ji}\}_{j \in N_i^c}\}$.

Summary

Optimal solution for one-shot decentralized estimation

Translating Radner's result

Since the model is a static team, from Radner's result we can say that the optimal estimates are

$$\hat{x}_i = F_i y_i$$

However, this form of the solution does not work well for the multi-step case.

An alternative form of the solution

Let $\hat{x}_i^{\mathrm{loc}} = \mathbb{E}[x \,|\, y_i]$. Then, the optimal estimates are given by

$$\boxed{\hat{x}_i = L_i \, \hat{x}_i^{\mathrm{loc}}, \quad L = -\Gamma^{-1}\eta}$$

- $L = \mathrm{vec}(L^1, \dots, L^n)$
- $\hat{\Sigma}_{ij} = \mathrm{cov}(\hat{x}_i, \hat{x}_j) = \Theta_i(\Sigma_{ii})^{-1}\Sigma_{ij}(\Sigma_{jj})^{-1}(\Theta_j)^\top$
- $\Gamma = [\Gamma_{ij}]$, where $\Gamma_{ij} = \hat{\Sigma}_{ij} \otimes R_{ij}$
- $\eta = \mathrm{vec}(P_1\hat{\Sigma}_{11}, \dots, P_n\hat{\Sigma}_{nn})$

Summary

Key observation The problem at time t is a one-shot optimization problem

Optimal estimator Let $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid I_i(t)]$ and $\hat{\Sigma}_{ij}(t) = \text{cov}(\hat{x}_i^{\text{loc}}(t), \hat{x}_j^{\text{loc}}(t))$. Then,

$$\boxed{\begin{aligned} \hat{x}_i(t) &= L_i(t)\, \hat{x}_i^{\text{loc}}(t), \\ \text{vec}(L_i(t)) &= -\left[\hat{\Sigma}_{ij}(t) \otimes R_{ij}\right]^{-1} \text{vec}(P_i \hat{\Sigma}_{ii}(t)) \end{aligned}}$$

Remarks To compute the optimal solution, we only need to compute $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ij}(t)$.

Recall, all random variables are jointly Gaussian. Pre-computing $\hat{\Sigma}_{ij}(t)$ and keeping track of $\hat{x}_i^{\text{loc}}(t)$ is trivial but for computational complexity.

Almost same as standard Kalman filtering! Relatively straight forward to come up with recursive equations (but for notation!).

Summary

Key observation The problem at time t is a one-shot optimization problem

Optimal estimator Let $\hat{x}_i^{\text{loc}}(t) = \mathbb{E}[x(t) \mid I_i(t)]$ and $\hat{\Sigma}_{ij}(t) = \text{cov}(\hat{x}_i^{\text{loc}}(t), \hat{x}_j^{\text{loc}}(t))$. Then,

$$\boxed{\begin{aligned} \hat{x}_i(t) &= L_i(t)\, \hat{x}_i^{\text{loc}}(t), \\ \text{vec}(L_i(t)) &= -\left[\hat{\Sigma}_{ij}(t) \otimes R_{ij}\right]^{-1} \text{vec}(P_i \hat{\Sigma}_{ii}(t)) \end{aligned}}$$

Remarks To compute the optimal solution, we only need to compute $\hat{x}_i^{\text{loc}}(t)$ and $\hat{\Sigma}_{ij}(t)$.

Recall, all random variables are jointly Gaussian. Pre-computing $\hat{\Sigma}_{ij}(t)$ and keeping track of $\hat{x}_i^{\text{loc}}(t)$ is trivial but for computational complexity.

Almost same as standard Kalman filtering! Relatively straight forward to come up with recursive equations (but for notation!).