

Lecture 28: Multiagent Consensus Equilibrium

Charles A. Bouman

Purdue University

Friday, October 30, 2026

Outline

- 1 From two agents to K heterogeneous agents
- 2 The weighted cost $f(x) = \sum_k \mu_k f_k(x)$ and its minimizer
- 3 The K proximal maps; stacked $\mathbf{w}$, $\mathbf{F}$, and $\mathbf{G}_\mu$
- 4 The MACE equations $\mathbf{F}(\mathbf{w}^*) = \mathbf{G}_\mu(\mathbf{w}^*)$
- 5 Proximal agents: the zero-gradient condition
- 6 General agents: equilibrium without a cost function
- 7 The role of the weights μ_k
- 8 Fixed point of $\mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})$, and one wrinkle
- 9 The MACE-DR algorithm

Core idea

Consensus equilibrium generalizes from two agents to K **weighted** agents. With proximal-map agents, the MACE solution minimizes the weighted-sum cost. With general agents, it is an equilibrium that balances the agents' actions.

Today's Big Ideas

- Multiagent consensus equilibrium solves an inverse problem with many heterogeneous agents, where some agents are proximal maps of cost functions and some are actions with no cost function.
- The MACE solution is the state at which every agent's output equals the same weighted average of the agents' inputs, and that weighted average is the reconstruction.
- Proximal-map agents built from proper closed convex costs make the MACE solution the minimizer of the weighted sum of those costs, while general agents give only a balance of the agents' actions.
- The weights set the relative strength of the agents, so the amount of regularization can be changed without retraining any agent.
- The weighted averaging operator is still idempotent, so the same Douglas-Rachford fixed-point iteration used for two agents also solves the MACE equations.

From two agents to many

We next consider the more general case in which we solve for the CE solution when there are **multiple heterogeneous agents**.

- Some of these agents will have the form of **proximal maps** corresponding to underlying objective functions.
- Some of them will be **directly implemented as operations performing actions**.

We refer to this problem as **multiagent consensus equilibrium (MACE)**.

In order to best understand MACE, we start by assuming the special case that there are K agents, each with a corresponding cost function, $f_k(x)$.

The weighted cost function

The complete cost function in this special case has the form

$$f(x) = \sum_{k=1}^K \mu_k f_k(x) , \quad (10.23)$$

where μ_k are weights such that $1 = \sum_{k=1}^K \mu_k$.

- Intuitively, the functions $f_k(x)$ may represent different soft or even hard constraints that one would like to incorporate into the solution of the inverse problem.

The desired solution to the problem is then given by

$$x^* = \arg \min_x \left\{ \sum_{k=1}^K \mu_k f_k(x) \right\} . \quad (10.24)$$

The K proximal maps

When each agent has a cost function, then each of the K agents must also have an associated proximal map given by

$$\begin{aligned} F_1(v) &= \arg \min_x \left\{ f_1(x) + \frac{1}{2\sigma^2} \|x - v\|^2 \right\} , \\ &\vdots \\ F_K(v) &= \arg \min_x \left\{ f_K(x) + \frac{1}{2\sigma^2} \|x - v\|^2 \right\} , \end{aligned}$$

where σ^2 is a **user-selectable parameter that does not affect the final solution**.

Stacked notation

Using the notation from the previous section, the stacked state and the stacked operator are given by

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}, \quad \mathbf{F}(\mathbf{w}) = \begin{bmatrix} F_1(w_1) \\ \vdots \\ F_K(w_K) \end{bmatrix}.$$

Finally, the stacked **weighted averaging operator** is defined as

$$\mathbf{G}_\mu(\mathbf{w}) = \begin{bmatrix} \bar{w} \\ \vdots \\ \bar{w} \end{bmatrix}, \quad \text{where} \quad \bar{w} = \sum_{k=1}^K \mu_k w_k.$$

The MACE equations

Using this notation, the MACE equations can now be written in the compact form

$$\mathbf{F}(\mathbf{w}^*) = \mathbf{G}_\mu(\mathbf{w}^*) , \quad (10.25)$$

where the solution is now given by

$$x^* = \sum_{k=1}^K \mu_k w_k^* .$$

- Every agent output equals the same weighted average $\bar{w}^*$.
- The reconstruction x^* is that weighted average.

The case of proximal-map agents

In the particular case when the agents are proximal maps, then by differentiating the MACE equations we can show that

$$\mathbf{0} \in \nabla f(x)|_{x=x^*} , \quad (10.26)$$

where x^* denotes the solution to the MACE equations.

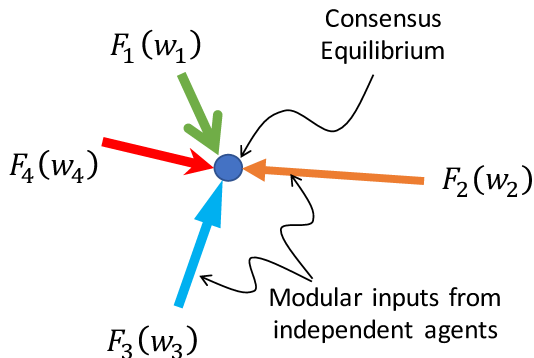
So if all the functions $f_k(x)$ are **proper closed convex** functions, then x^* is a MACE solution **if and only if** it is the minimum of $f(x)$.

General agents: equilibrium without a cost function

However, the power of the MACE equations is when we use them with **plug-in agents F_k that are not proximal maps.**

- In this case, the solution to the MACE equations does **not** correspond to minimizing any underlying cost function.
- But it does represent an **equilibrium that balances the agents' actions.**

A graphical interpretation of the MACE solution



Each agent is depicted as an action that moves the state from a starting point of w_k to the common central point of x^* that constitutes the MACE solution. We also have the additional constraint that $x^* = \sum_{k=1}^K \mu_k F_k(w_k)$.

The solution falls at the point at which the agents' forces are in balance, but this does not necessarily correspond to the minimization of a cost function.

The weights μ_k

The weights μ_k represent an important and very useful generalization of the CE method.

- By using these weights, it is very easy to adjust the **relative strength** of the different agents by simply adjusting the weighted average used in computing $\mathbf{G}_\mu$.
- For example, this can be used to adjust the **regularization** by adjusting the relative weighting of the forward and prior agents.

Why this matters

Adjusting these weights does not require that one change the agents, F_k . This is important since if the agents are designed using machine learning methods, they may require computationally expensive **retraining** every time they are adjusted.

The same fixed-point machinery still applies

Regardless of the particular choice of weights, the averaging operator $\mathbf{G}_\mu$ still has the critical property that

$$\mathbf{G}_\mu \mathbf{G}_\mu = \mathbf{G}_\mu .$$

Therefore, we once again know that the solution to the CE equations is given by

$$\mathbf{T}\mathbf{w}^* = \mathbf{w}^* , \quad \text{where} \quad \mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I}) .$$

- Again, we can solve for the fixed point of $\mathbf{T}$ using the DR algorithm with Mann iterations.
- We know that this algorithm will converge when $\mathbf{T}$ is nonexpansive.

One wrinkle in the story

- When $\mu_k = 1/K$, then $\mathbf{G}_\mu$ is a **projection**, and $(2\mathbf{G} - I)$ is firmly nonexpansive.
- However, when this is not the case, then $\mathbf{G}_\mu$ may be **expansive** (see Chapter Problems).
- In this case, $(2\mathbf{G} - I)(2\mathbf{F} - I)$ may not be nonexpansive, and the MACE-DR algorithm may, theoretically speaking, **not converge**.

In practice, this never seems to happen, but it is an interesting point to keep in mind.

Pseudocode: the MACE-DR algorithm

MACE-DR Algorithm

Initialize Reconstruction: $x^{(0)} \in \mathbb{R}^N$

$$\mathbf{w} \leftarrow \begin{bmatrix} x^{(0)} \\ \vdots \\ x^{(0)} \end{bmatrix}$$

Until Converged {

$$\mathbf{x} \leftarrow \mathbf{F}(\mathbf{w})$$

$$\mathbf{z} \leftarrow \mathbf{G}_\mu(2\mathbf{x} - \mathbf{w})$$

$$\mathbf{w} \leftarrow \mathbf{w} + 2\rho(\mathbf{z} - \mathbf{x})$$

}

$$x^* \leftarrow \mathbf{z}_1$$

Return(x^*)

Solves the CE equations for multiple plug-in agents $F_k(x)$ using DR with Mann iterations. Provably convergent when $\mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})$ is nonexpansive.

What the algorithm is doing, and what it buys us

The book's picture

“Intuitively, you can imagine these functions as anthropomorphic agents, each of which has the assignment to move the solution in a way that solves ‘their’ problem. Much like the endless bureaucracy in any large organization, each MACE agent may attempt to enforce its preference, which might be in conflict with the objectives or actions of other agents. The CE solution is the solution that achieves an equilibrium in the actions of all these agents.”

- In the special case that the agents F_k are proximal maps, the algorithm converges to x^* , the solution of (10.24).
- The real power of this method is that it works with **any combination** of agents that are proximal maps (objective models) or not proximal maps (action models), and it converges, when $\mathbf{T}$ is nonexpansive, to a solution that **balances all these agents' forces**.

Summary

- MACE solves for a consensus equilibrium among K heterogeneous agents F_k with weights μ_k satisfying $1 = \sum_{k=1}^K \mu_k$, where some agents are proximal maps and some are actions.
- The MACE equations are $\mathbf{F}(\mathbf{w}^*) = \mathbf{G}_\mu(\mathbf{w}^*)$, where $\mathbf{G}_\mu$ replaces every entry by $\bar{w} = \sum_{k=1}^K \mu_k w_k$, and the reconstruction is $x^* = \sum_{k=1}^K \mu_k w_k^*$.
- When each F_k is the proximal map of a proper closed convex f_k , the solution satisfies $\mathbf{0} \in \nabla f(x)|_{x=x^*}$ for $f(x) = \sum_{k=1}^K \mu_k f_k(x)$, so $x^* = \arg \min_x f(x)$, while general agents give an equilibrium with no cost function.
- The weights μ_k set the relative strength of the agents, so the regularization can be changed without retraining any agent.
- Since $\mathbf{G}_\mu \mathbf{G}_\mu = \mathbf{G}_\mu$, the solution is the fixed point of $\mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})$, computed by MACE-DR with Mann iterations.

Reading: Section 10.4. Next class (Nov. 2): Chapter 12, the EM algorithm.