6.S979 Topics in Deployable Machine Learning
Lecture: Decentralized Optimization, Decision Making and Control

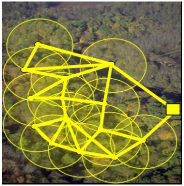
Asu Ozdaglar Pablo A. Parrilo
MIT

October 3, 2019

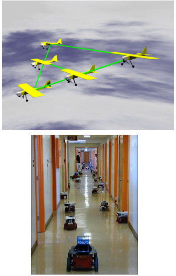
Motivation

- Many modern systems are large-scale, consist of agents with local information and involve collection and processing of data in a decentralized manner.
- This motivated much interest in developing distributed algorithms for processing of large-scale data, and control and optimization of multi-agent networked systems.

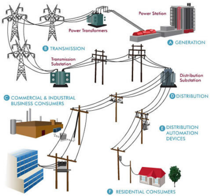
Routing and congestion control in wireline and wireless networks



Parameter estimation in sensor networks



Multi-agent cooperative control



Smart grid systems

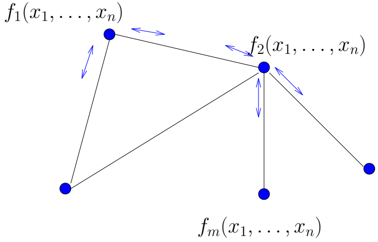
Distributed Multi-agent Optimization

- Many of these problems can be represented within the general formulation:
- A set of agents (nodes) $\{1, \ldots, N\}$ connected through a network.

- The goal is to cooperatively solve

$$\begin{array}{ll} \min\limits_{x} & \sum_{i=1}^{N} f_i(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array}$$

$f_i(x) : \mathbb{R}^n \to \mathbb{R}$ is a convex (possibly nonsmooth) function, known only to agent i.



- Since such systems often lack a centralized processing unit, algorithms for this problem should involve each agent performing computations locally and communicating this information according to the underlying network.

Machine Learning Example

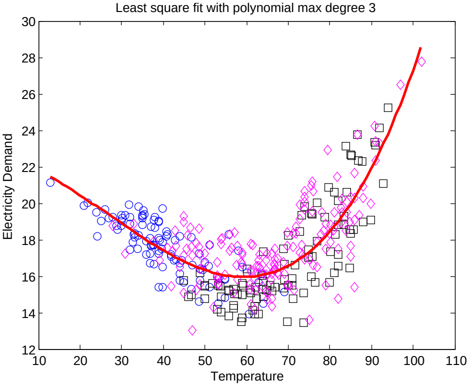
- A network of 3 sensors.
- Data is collected at different sensors: temperature t, electricity demand d.
- System goal: learn a degree 3 polynomial electricity demand model:

$$d(t) = x_3 t^3 + x_2 t^2 + x_1 t + x_0.$$

- System objective:

$$\min_x \quad \sum_{i=1}^{3} ||A_i' x - d_i||_2^2 .$$

where $A_i = [1, t_i, t_i^2, t_i^3]'$ at input data t_i.



Machine Learning General Set-up

- A network of agents $i = 1, \ldots, N$.
- Each agent i has access to local feature vectors A_i and output b_i.
- System objective: train weight vector x to

$$\min_{x} \quad \sum_{i=1}^{N} L(A_i' x - b_i) + p(x),$$

for some loss function L (on the prediction error) and penalty function p (on the complexity of the model).

  - Example: Least-Absolute Shrinkage and Selection Operator (LASSO):

$$\min_{x} \quad \sum_{i=1}^{N} ||A_i' x - b_i||_2^2 + \lambda \, ||x||_1 \, .$$

Literature: Parallel and Distributed Optimization

- Lagrangian relaxation and dual optimization methods:
  - Dual gradient ascent, (single) coordinate ascent methods.
- Parallel computation and optimization:
  - [Tsitsiklis 84], [Bertsekas and Tsitsiklis 95].
- Consensus and cooperative control:
  - Averaging algorithms: Deterministic averaging of all neighbor estimates.
    [Jadbabaie, Lin, and Morse 03], [Olfati-Saber and Murray 04], [Olshevsky and Tsitsiklis 07], [Tahbaz-Salehi and Jadbabaie 08], [Kar and Moura 09], [Frasca, Carli, Fagnani and Zampieri 09], [Bullo, Cortes, Martinez 09],[Oreshkin, Coates, and Rabbat 10].
  - Gossip algorithms: Random pairwise averaging.
    [Boyd, Ghosh, Prabhakar, and Shah 05], [Dimakis, Sarwate, and Wainwright 08], [Fagnani, Zampieri 09], [Aysal, Yildiz, Sarwate, and Scaglione 09].

Literature: Distributed Multi-agent Optimization

- Distributed first order primal subgradient methods [Nedic, Ozdaglar 09].
- Various extensions:
  - Local and global constraints [Nedic, Ozdaglar, Parrilo 10], [Zhu and Martinez 10].
  - Randomly varying communication networks[Lobel, Ozdaglar 09], [Baras and Matei 10], [Lobel, Ozdaglar, and Feijer 10].
  - Network effects [Nedic, Olshevsky, Ozdaglar, Tsitsiklis 09]
  - Random gradient errors [Ram, Nedic, Veeravalli 09].
- Ordinary-Augmented Lagrangian primal-dual subgradient methods
  - [Jakovetic, Xavier, Moura 11], [Zhu, Giannakakis, Cano 09],[Mota, Xavier, Aguiar, Puschel 13]
- Distributed second order methods (for more specialized problems)
  - [Wei, Ozdaglar, Jadbabaie 11], [Liu, Sherali 12]

This Lecture

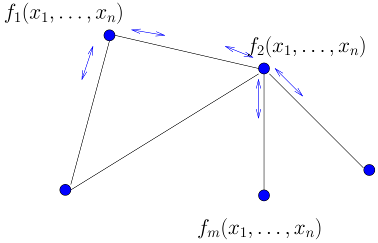
- Brief overview of distributed primal subgradient methods [Nedic, Ozdaglar 09].
- Other distributed optimization methods.
- Decentralized strategic decision making.
- Decentralized control.

Distributed Subgradient Method

- Recall problem formulation:

$$\min_{x} \quad \sum_{i=1}^{N} f_i(x)$$
$$\text{s.t.} \quad x \in \mathbb{R}^n$$

f^*: optimal value.



- We assume agents are connected through a "time-varying" graph.
- Key idea: Each agent maintains a local estimate of the optimal solution, and updates it by taking a (sub)gradient step along his local objective function and averaging with neighbors' estimates.

Distributed Subgradient Method

- Let $x^i(k) \in \mathbb{R}^n$ denote agent i's estimate of the solution at time k.

Agent Update Rule:

- At each time k, agent i updates its estimate as:

$$x_i(k+1) = \sum_{j=1}^{N} a_{ij}(k)x_j(k) - \alpha(k)d_i(k),$$

  $a_{ij}(k) \geq 0$: weights, $\alpha(k) > 0$: stepsize, $d_i(k)$: a subgradient of f_i at $x_i(k)$.

- The weights $a_{ij}(k)$ represents i's time-varying neighbors at time k: $a_{ij}(k) > 0$ only for agent j that communicate with agent i at time k.
- When all $f_i = 0$, the method reduces to the consensus algorithm [Vicsek 95], [Jadbabaie, Lin, Morse 03].

Linear Dynamics and Transition Matrices

- We let $A(k)$ denote the weight matrix $[a_{ij}(k)]_{i,j=1,\ldots,N}$, and define transition matrices

$$\Phi(k,s) = A(k)A(k-1)\cdots A(s+1)A(s) \qquad \text{for all } k \geq s$$

- We use these matrices to relate $x_i(k+1)$ to $x_j(s)$ at time $s \leq k$:

$$x_i(k+1) \;=\; \sum_{j=1}^{N}[\Phi(k,s)]_{ij}x_j(s) - \sum_{r=s}^{k-1}\sum_{j=1}^{N}[\Phi(k,r+1)]_{ij}\alpha(r)d_j(r) - \alpha(k)\,d_i(k).$$

- We analyze convergence properties of the distributed method by establishing:
  - Convergence of transition matrices $\Phi(k,s)$ (consensus part)
  - Convergence of an approximate subgradient method (effect of optimization)

Assumptions: Weights and Connectivity

Assumption (Weights)

(a) *There exists a scalar* $\eta \in (0,1)$ *s.t.* $a_{ii}(k) \geq \eta$ *and if* $a_{ij}(k) > 0$, $a_{ij}(k) \geq \eta$.

(b) *The weight matrix* $A(k)$ *is doubly stochastic,* $\sum_{j=1}^{N} a_{ij}(k) = 1$ *for all* i *and* $\sum_{i=1}^{N} a_{ij}(k) = 1$ *for all* j.

- Double stochasticity ensures agent estimates equally influential in the limit. This guarantees minimizing the sum of the local objective functions.
- Represent information exchange by (V, E_k),

$$E_k = \{(j,i) \mid a_{ij}(k) > 0,\ i,j = 1,\ldots,m\}.$$

Assumption (Connectivity)

There exists an integer $B \geq 1$ *such that the directed graph*

$$\left(\mathcal{M}, E_k \cup \cdots \cup E_{k+B-1}\right)$$

is strongly connected for all $k \geq 0$.

Convergence Analysis – Idea

- Recall the evolution of the estimates (with $\alpha(s) = \alpha$):

$$x_i(k+1) \quad = \quad \sum_{j=1}^{N} [\Phi(k,s)]_{ij} x_j(s) - \alpha \sum_{r=s}^{k-1} \sum_{j=1}^{N} [\Phi(k, r+1)]_{ij} d_j(r) - \alpha d_i(k).$$

- Proof method: We define an auxiliary sequence: $y(k) = \frac{1}{N} \sum_{i=1}^{N} x_i(k)$.
- The sequence $y(k)$ evolves as

$$y(k+1) = y(k) - \frac{\alpha}{N} \sum_{i=1}^{N} d_i(k),$$

where $d_i(k)$ is a subgradient of f_i at $x_i(k)$.

- This corresponds to an approximate subgradient method for minimizing $\sum_j f_j(x)$ (subgradients computed at $x_i(k)$ instead of $y(k)$).

Convergence Analysis – Idea

- But $y(k)$ evolution can be written as:

$$y(k+1) = \frac{1}{N}\sum_{j=1}^{N} x_j(s) - \frac{\alpha}{N}\sum_{r=s}^{k-1}\sum_{j=1}^{N} d_j(r) - \frac{\alpha}{N}\sum_{i=1}^{N} d_i(k).$$

- Using the below result, this shows that $y(k)$ and $x_i(k)$ get close to each other in the limit: agent "disagreements" disappear and the method behaves as a centralized subgradient method.

Theorem (Nedic, Olshevsky, Ozdaglar, Tsitsiklis 09)

For all i, j and all k, s with $k \geq s$, we have

$$\left|[\Phi(k,s)]_{ij} - \frac{1}{N}\right| \leq \left(1 - \frac{\eta}{4N^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2}.$$

Convergence

- We assume set of subgradients of f_i uniformly bounded by some $L > 0$.
- Let $\hat{x}_i(k) = \frac{1}{k}\sum_{h=1}^{k} x_i(h)$: ergodic average of estimates.

Proposition

For all $k \geq 1$,

$$f(\hat{x}_i(k)) \leq f^* + \frac{\alpha L^2 C}{2} + \frac{m}{2\alpha k}\, dist(y(0), X^*),$$

where $\beta = 1 - \frac{\eta}{4N^2}$ *and* $C = 1 + 8N\left(2 + \frac{NB}{\beta(1-\beta)}\right)$.

- With constant stepsize, this achieves:

$$\limsup_{k\to\infty} f(\hat{x}_i(k)) \leq f^* + \frac{\alpha L^2 C}{2} \qquad \text{for all } i.$$

- By choosing $\alpha(k) = 1/\sqrt{k}$, this achieves a convergence rate of $O(1/\sqrt{k})$.

Other Distributed Optimization Methods

- We can also use Alternating Direction Method of Multipliers (ADMM)-type methods for distributed optimization.
  - Involves reformulation into a separable problem and sequential updates of subcomponents of the decision vector.
- Introduce a "local copy" x_i in $\mathbb{R}^n$ for each i and write

$$\begin{aligned} \min_{x \in \mathbb{R}^{mn}} \quad & \sum_{i=1}^{m} f_i(x_i) \\ s.t. \quad & (1) \text{ or } (2). \end{aligned}$$

(1) Edge-based reformulation: $x_i = x_j$ for $(i,j) \in E$.

(2) Node-based reformulation: $x_i = \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} x_j$ for $i \in V$

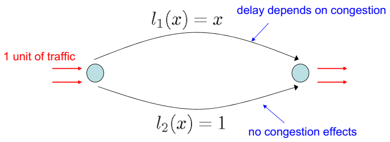
- Rate guarantees for the convex and strongly convex/smooth case [Makhdoumi, Ozdaglar 15].

Other Distributed Optimization Methods

- Standard distributed gradient method [Nedic, Ozdaglar 09].
  - [Yuan, Lin, Yin 16] considered this algorithm for when the local functions are smooth and when they are convex or strongly convex.
  - For the convex case, they show the network-wide mean estimate converges at rate $O(1/k)$ to an $O(\alpha)$ neighborhood of the optimal solution, and for the strongly convex case, all local estimates converge at a linear rate $O(\exp(-k/\Theta(\kappa)))$ to an $O(\alpha)$ neighborhood of the optimal solution (κ is the condition number).
- Extra: [Shi, Ling, Wu, Yin 15] provides a novel algorithm which can be viewed as a primal-dual algorithm for the constrained reformulation of the problem.
- Gradient Tracking: [Qu and Li 18] proposes to update the DG method such that agents also maintain, exchange, and combine estimates of gradients of the local objective functions.
- See [Jakovetic 19] for a unified analysis of these methods, and [Fallah et al. 19] for accelerated and noisy versions of these algorithms.

From Distributed Optimization to Games

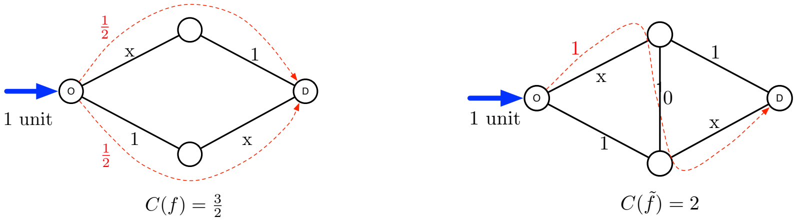
- Early 2000: Resource allocation among strategic/self-interested agents.
- Selfish Routing [Roughgarden, Tardos 00]
  - Source-based routing in communication networks, efficiency of traffic flows in transport systems.
  - Price of Anarchy: quantification of efficiency losses



- Service Provider Incentives in Traffic Engineering
  - Pricing and Efficiency in Congested Markets [Acemoglu, Ozdaglar 07].
  - Partially Optimal Routing (optimal routing within subnetworks is overlaid with selfish routing across domains) [Acemoglu, Johari, Ozdaglar 07].

From Distributed Optimization to Games

- Paradoxes of strategic decision making:



- Information and Learning in Traffic Networks
  - Effect of information in congested traffic [Acemoglu, Makhdoumi, Malekian, Ozdaglar 17].
  - Information Design!

Games and equilibria

- Multiple decision makers, with possibly competing goals
- A common model for strategic interactions among different agents
- Two-player, zero-sum case is very special (minimax theorem).
- General case requires a different solution concept: Nash equilibrium

Games and equilibria

- Finite games in normal (strategic) form:

$$\mathcal{G} = \langle \mathcal{M}, \{E^m\}_{m\in\mathcal{M}}, \{u^m\}_{m\in\mathcal{M}} \rangle$$

  where

  - $\mathcal{M}$ is the set of players
  - E^m are the possible strategies of player m
  - u^m is the utility (payoff) of player m

- Players choose their actions simultaneously and independently
- Typically, may need to consider mixed strategies, where players randomize among possible actions according to specific probabilities

Nash equilibria

- Natural solution concept, extends usual minimax (zero-sum games)
- Key idea: No player should benefit from unilateral deviations

Definition

A strategy profile $p = \{p_1, \dots, p_M\}$ is a Nash equilibrium if

$$u^m(p^m, p^{-m}) \geq u^m(q^m, p^{-m})$$

for all players $m \in \mathcal{M}$ and every strategy $q^m \in E^m$.

- Nash equilibria always exist
- May not be unique.
- May require mixed strategies

Potential Games

A nice class of games, with appealing mathematical properties

- $\mathcal{G}$ is an exact potential game if $\exists\, \Phi : E \to \mathbb{R}$ ("potential") such that

$$\Phi(x^m, x^{-m}) - \Phi(y^m, x^{-m}) = u^m(x^m, x^{-m}) - u^m(y^m, x^{-m}),$$

- Weaker notion: ordinal potential game, if the utility differences above agree only in sign.
- Potential Φ aggregates and explains incentives of all players.
- Examples: congestion games, etc.

In potential games, finding equilibria reduces to optimization!

Potential Games

A nice class of games, with appealing mathematical properties

- $\mathcal{G}$ is an exact potential game if $\exists\, \Phi : E \to \mathbb{R}$ ("potential") such that

$$\Phi(x^m, x^{-m}) - \Phi(y^m, x^{-m}) = u^m(x^m, x^{-m}) - u^m(y^m, x^{-m}),$$

- Weaker notion: ordinal potential game, if the utility differences above agree only in sign.
- Potential Φ aggregates and explains incentives of all players.
- Examples: congestion games, etc.

In potential games, finding equilibria reduces to optimization!

Potential Games and Learning

- A global maximum of an ordinal potential is a pure Nash equilibrium.
- Every finite potential game has a pure equilibrium.
- Many decentralized learning dynamics (such as better-reply dynamics, fictitious play, spatial adaptive play) "converge" to a pure Nash equilibrium [Monderer and Shapley 96], [Young 98], [Marden, Arslan, Shamma 06, 07].

Potential Games

- When is a given game a potential game?
- More important, what are the obstructions, and what is the underlying structure?
- Can we "approximate" general games with potential games?
- Geometric characterization, connections to Helmholtz decomposition [Candogan et. al 11]
- Convergence of learning dynamics in near-potential games [Candogan, Ozdaglar, Parrilo 13]