LECTURE #1: Motivation & Examples

Professor Ali H. Sayed
UCLA Electrical Engineering


Pre-Requisites

Prior training expected in:

**Probability theory:** scalar and vector random variables, pdf, mean, variance, correlation, independence.

**Linear algebra:** matrices, inverses, range and null-spaces, rank, vector and matrix norms, Kronecker products, linear systems of equations, eigen-decomposition, singular-value decomposition, Jordan decomposition.
Organization

The course consists largely of five parts:

1. **Background Material**: Linear Algebra and Matrix Theory Results, Complex Gradients and Complex Hessian Matrices, Convexity, Strict Convexity, and Strong Convexity, Mean-Value Theorems, Lipschitz Conditions.


Organization

Motivation#1

Course deals with the topic of information processing over graphs.

→ Multi-agent networks for the distributed solution of adaptation, learning, and optimization problems from streaming data through localized interactions.

Internet map (2005). Wikimedia commons.
Motivation#2

The results derived here are useful in:

- Comparing network configurations against each other;
- Comparing networks against batch solutions;
- Understanding limits of performance;
- Understanding benefits & pitfalls of cooperation;
- Highlighting interesting phenomena over networks.
Motivation#3

Motivation#4

We will examine the analysis and design of networked solutions plus applications in:

- distributed sensing
- intrusion detection
- distributed estimation
- online learning
- pattern classification
- clustering
- distributed optimization
- multi-agent systems
Concepts & Examples
Deals with the discovery of **global** information from **local** interactions among **dispersed** agents.

**Features:**
- Common objective(s);
- In-network processing;
- Dispersed agents.
Centralized Processing

→ Exchange of data between the dispersed agents and a **fusion** center.

- Cost of communications;
- Privacy & security considerations;
- Critical point of failure.
Why Distributed Processing?

- Data already available at dispersed locations (cloud).
- Power of cooperation → mining of Big Data sets.
- Privacy and security considerations.
- Robustness and resilience (biological networks).
- Robotic swarms (disaster areas).
- Network science (social networks).

*Source: IEEE SPM May 2013*
Nature provides splendid examples of real-time decentralized learning & adaptation.
One Useful Result

Consensus Construction (1974)

Each agent $k$ has a measurement $x_k$. 

**Objective:** Compute average value.

$$x_3 \leftarrow \sum_{\ell \in N_3} a_{\ell3} \ x_\ell$$

$$x_k \rightarrow \frac{1}{N} \sum_{\ell=1}^{N} x_\ell$$
One Useful Application

Technique used in **The Lion King** (1994) and **Batman Returns** (1992) to produce swarming effects.

YouTube: [https://www.youtube.com/watch?v=2m-42ek85G4](https://www.youtube.com/watch?v=2m-42ek85G4)
Issue#1: Cognition

Biological networks have more complex objectives such as tracking food sources or evading predators.

Source: http://youtu.be/zvfY8-3ktNA
Issue #2: Interactions

Interactions are information-aware \(\rightarrow\) informed vs. uninformed agents

Source: Collective Animal Behavior Lab (I. D. Couzin, Princeton University)
Adaptive Networks

- **Adaptive agents:** learn from streaming data.
- **Cooperative agents:** interact locally.
- **Adaptive topology:** re-wire the graph.
- **Distributed optimization:** solve meaningful problems.

\[
\min_w \sum_{k=1}^{N} J_k(w) \\
\text{subject to } \begin{cases} 
g_k(w) \leq 0 \\
h_k(w) = 0 \end{cases}
\]
Example #1: Cooperation

Emulating fish schooling and prey-predator behavior → Cooperative Networks.
Example#2: Competition

Emulating prey-predator behavior $\rightarrow$ Network Competition
Example #3: Cognitive Radios

- Secondary user
- Primary user

Swarming secondary users

PSD (mW/Hz)

Frequency (Mhz)
Example #4: Optimization

\[
\begin{align*}
\min_w \sum_{k=1}^{N} J_k(w) \\
\text{subject to} \quad \left\{ \begin{array}{l}
g_k(w) \leq 0 \\
h_k(w) = 0
\end{array} \right.
\end{align*}
\]
Example#5: Dictionary Learning

Sparse representation of a signal using atoms from a dictionary,

\[ \mathbf{x}_t \in \mathbb{R}^M \]

\[ W \in \mathbb{R}^{M \times K} \]

\[ \mathbf{y}_i^c \]

private dictionaries

private opinions
Example #5: Dictionary Learning

\[
\mathbf{W} = \begin{bmatrix}
W_1 & W_2 & \ldots & W_N
\end{bmatrix}
\]

\[
\mathbf{y}_t^o = \text{col}\{\mathbf{y}_{1,t}^o, \mathbf{y}_{2,t}^o, \ldots, \mathbf{y}_{N,t}^o\}
\]

\[
\mathbf{x}_t \approx \sum_{k=1}^{N} W_k \mathbf{y}_{k,t}^o
\]
Relevant Questions

- What strategies enable distributed learning?
- How to ensure stable behavior?
- What are the limits of performance?
- Can we match centralized processing?
- Does cooperation always help?
- Does it help to have more agents?
End of Lecture