ECE 286 Class 15: Graph-Based Cooperative Localization

Florian Meyer

Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Graph-Based Cooperative Localization

- With each agent k, there is associated a time-dependent "agent state" vector $\mathbf{x}_{k,n}$, $n = 1, 2, \ldots$ (e.g., time-dependent location of agent)
- Each agent k acquires time-dependent pairwise measurements $\mathbf{y}_{kl,n}$ involving other agents $l \in \mathcal{N}_{k,n}$
- Each agent k aims to estimate its state $\mathbf{x}_{k,n}$ from $\mathbf{y}_{1:n}$ (i.e., from all the measurements $\mathbf{y}_{k'l,n'}$ for $k' = 1, \ldots, K$, $l \in \mathcal{N}_{k',n}$, $n' = 1, \ldots, n$)
- Fully distributed: no fusion center, only local communications

[**Wymeersch et al., 09**] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, 2009.

[**Win et al., 18**] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, "Efficient multi-sensor localization for the Internet-of-Things," *IEEE Signal Process. Mag.*, 2018.

System Model

State-transition model for agent k

$$\mathbf{x}_{k,n} = \mathbf{g}_{k,n}(\mathbf{x}_{k,n-1}, \underbrace{\mathbf{u}_{k,n}}_{\text{Driving noise (white)}}) , \quad n = 1, 2, \ldots$$

Measurement model at agent k

$$\mathbf{y}_{kl,n} = \mathbf{h}_{kl,n}(\mathbf{x}_{k,n}, \mathbf{x}_{l,n}, \underbrace{\mathbf{v}_{kl,n}}_{\text{Measurement noise (white)}}) , \quad l \in \mathcal{N}_{k,n}, \quad n = 1, 2, \ldots$$

- Driving noise is statistical independent across agents k
- Measurement noise is statistical independent across edges (k, l)
- Driving and measurement noise are mutually independent and independent of initial states $\mathbf{x}_{k,0}$, $k = 1, \ldots, K$

Problem Formulation

State-transition model for agent k

$$\mathbf{x}_{k,n} = \mathbf{g}_{k,n}(\mathbf{x}_{k,n-1}, \underbrace{\mathbf{u}_{k,n}}_{\text{Driving noise (white)}}), \quad n = 1, 2, \dots$$

Measurement model at agent k

$$\mathbf{y}_{kl,n} = \mathbf{h}_{kl,n}(\mathbf{x}_{k,n}, \mathbf{x}_{l,n}, \underbrace{\mathbf{v}_{kl,n}}_{\text{Measurement noise (white)}}), \quad l \in \mathcal{N}_{k,n}, \quad n = 1, 2, \dots$$

- MMSE estimator:

$$\hat{\mathbf{x}}_{k,n} = \mathrm{E}\{\mathbf{x}_{k,n}|\mathbf{y}_{1:n}\} = \int \mathbf{x}_{k,n} \, f(\mathbf{x}_{k,n}|\mathbf{y}_{1:n}) \, \mathrm{d}\mathbf{x}_{k,n}$$

- The posterior pdf $f(\mathbf{x}_{k,n}|\mathbf{y}_{1:n})$ can be calculated at agent k by a distributed and "dynamic" sum-product algorithm (SPA)

Derivation of the Factor Graph

- Consider the joint state $\mathbf{x}_n \triangleq (\mathbf{x}_{1,n}^{\mathrm{T}} \cdots \mathbf{x}_{K,n}^{\mathrm{T}})$

- Joint prior distribution:

$$f(\mathbf{x}_{0:n}) = \left(\prod_{l=1}^{K} f(\mathbf{x}_{l,0}) \right) \prod_{n'=1}^{n} \prod_{k=1}^{K} f(\mathbf{x}_{k,n'} | \mathbf{x}_{k,n'-1})$$

- Joint likelihood function:

$$f(\mathbf{y}_{1:n} | \mathbf{x}_{1:n}) = \prod_{n'=1}^{n} \prod_{(k,l) \in \mathcal{E}_{n'}} f(\mathbf{y}_{kl,n} | \mathbf{x}_{k,n'}, \mathbf{x}_{l,n'})$$

Derivation of the Factor Graph

- The posterior pdf $f(\mathbf{x}_{k,n}|\mathbf{y}_{1:n})$ is obtained by marginalizing the joint posterior pdf $f(\mathbf{x}_{0:n}|\mathbf{y}_{1:n})$:

$$f(\mathbf{x}_{k,n}|\mathbf{y}_{1:n}) = \int f(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}) \mathrm{d}\mathbf{x}_{\sim k,n}$$

- Factorization of the joint posterior pdf:

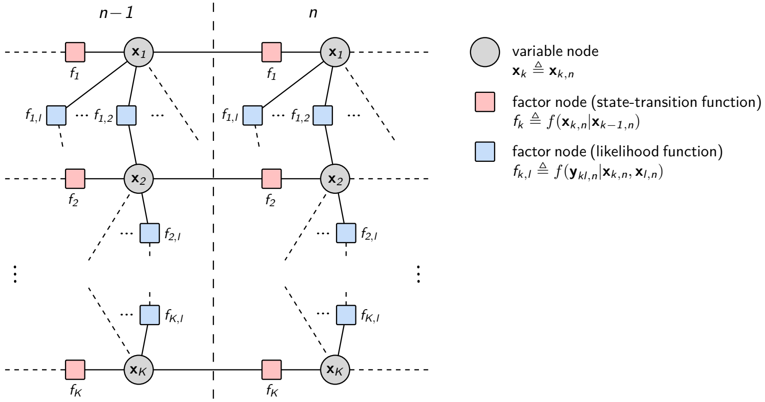
$$\begin{aligned} f(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}) &\propto f(\mathbf{x}_{0:n}) f(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}) \\ &= \left(\prod_{l=1}^{K} f(\mathbf{x}_{l,0})\right) \prod_{n'=1}^{n} \left(\prod_{k=1}^{K} f(\mathbf{x}_{k,n'}|\mathbf{x}_{k,n'-1})\right) \\ &\qquad \times \prod_{(k',l')\in\mathcal{E}_{n'}} f(\mathbf{y}_{k'l',n}|\mathbf{x}_{k',n'}, \mathbf{x}_{l',n'}) \end{aligned}$$

The Factor Graph

- Recall factorization:

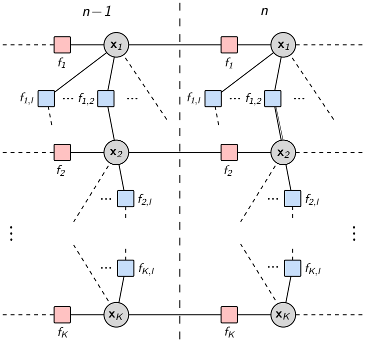
$$f(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}) \propto \Big(\prod_{l=1}^{K} f(\mathbf{x}_{l,0})\Big) \prod_{n'=1}^{n} \Big(\prod_{k=1}^{K} f(\mathbf{x}_{k,n'}|\mathbf{x}_{k,n'-1})\Big) \prod_{(k',l')\in\mathcal{E}_{n'}} f(\mathbf{y}_{k'l',n'}|\mathbf{x}_{k',n'},\mathbf{x}_{l',n'})$$

- Representation by factor graph:



The Factor Graph

- Factor graph:



- Problem: Factor graph grows with time $\Rightarrow$ computation, communication, and memory requirements increase linearly with time
- Solution: Messages are sent only forward in time and iterative message passing is performed at each time step individually

Message Passing

Dynamic SPA algorithm:

- "Prediction" message:

$$\phi_{\rightarrow n}(\mathbf{x}_{k,n}) = \int f(\mathbf{x}_{k,n}|\mathbf{x}_{k,n-1})\, b(\mathbf{x}_{k,n-1})\, \mathrm{d}\mathbf{x}_{k,n-1}$$

  Since we send messages only forward in time, we directly use the belief $b(\mathbf{x}_{k,n-1})$ instead of some extrinsic information

- "Measurement" message:

$$\phi_{l\rightarrow k}(\mathbf{x}_{k,n}) = \int f(\mathbf{y}_{kl,n}|\mathbf{x}_{l,n},\mathbf{x}_{k,n})\, \psi_{l\rightarrow k}(\mathbf{x}_{l,n})\, \mathrm{d}\mathbf{x}_{l,n}$$

- Extrinsic information:

$$\psi_{l\rightarrow k}(\mathbf{x}_{l,n}) = \phi_{\rightarrow n}(\mathbf{x}_{l,n}) \prod_{k' \in \mathcal{N}_{l,n}\backslash\{k\}} \phi_{k'\rightarrow l}(\mathbf{x}_{l,n})$$

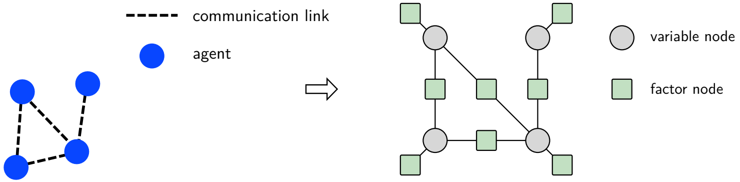
- Belief:

$$b(\mathbf{x}_{k,n}) \propto \phi_{\rightarrow n}(\mathbf{x}_{k,n}) \prod_{l \in \mathcal{N}_{k,n}} \phi_{l\rightarrow k}(\mathbf{x}_{k,n})$$

Distributed Implementation

- A **distributed** implementation of the dynamic SPA algorithm presupposes that the communication graph of the agent network coincides with the factor graph

  - Agent k in the communication graph corresponds to variable node $\mathbf{x}_{k,n}$ in the factor graph
  - Agent k is able to communicate with all neighboring agents $l \in \mathcal{N}_{k,n}$



- This correspondence guarantees that all the messages required for calculating the belief $b(\mathbf{x}_{k,n})$ at agent k are within the "communication neighborhood" of agent k

Message Representation

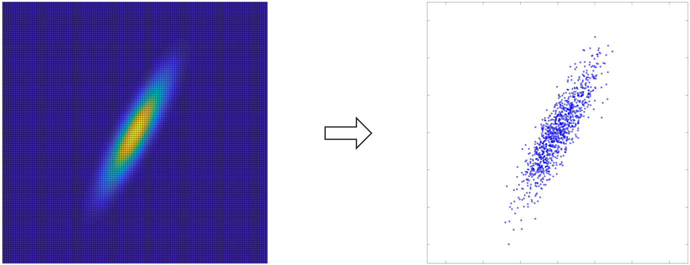
- Direct implementation of the SPA algorithm (message and belief calculation rules) is still computationally infeasible
- Two alternative feasible approximations:
  - using a parametric representation for the messages and beliefs
    $\Rightarrow$ Gaussian SPA, ...
  - using a particle representation for the messages and beliefs
    $\Rightarrow$ nonparametric SPA, ...

Parametric Representation

- Messages and beliefs are represented by parameters of a suitable parametric distribution
- In linear-Gaussian systems, all messages and beliefs are Gaussian and can thus be represented by a mean and a covariance matrix $\Rightarrow$ Gaussian SPA
- In nonlinear and/or non-Gaussian systems, parametric distributions tailored to the underlying problem have to be used (e.g., mixture of Gaussians, annularly shaped, spherically shaped)
- Low computation and communication requirements

Particle Representation / Nonparametric SPA

- Each message or belief is represented by a large number of particles and weights: $f(\mathbf{x}) \sim \left\{ \left(\mathbf{x}^{(j)}, w^{(j)} \right) \right\}_{j=1}^{J}$



- Nonparametric SPA uses a particle representation and is suited to arbitrary nonlinear, non-Gaussian systems

[**Ihler et al., 05**] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, 2005.