Marginal Density Averaging for Distributed Node Localization from Local Edge Measurements

Parth Paritosh, Nikolay Atanasov, and Sonia Martinez

***Abstract*— The cooperative localization problem consists of a group of networked agents aiming to find the true probability density function(pdf) of their states. Unlike existing algorithms such as Distributed Kalman filters or non-Bayesian social learning, our algorithm restricts each agent's estimates to a local pdf on its own and its neighbors' state variables. The agents update these pdfs via local observations of their neighbors and their shared messages. This partial state estimation problem is formulated as a distributed constrained optimization in the space of probability density functions. Consistent estimates across the agents are enforced with a constraint requiring equal estimated densities over common states in every communicating agent pair. Stochastic mirror descent steps are then computed to develop a novel cooperative estimation algorithm with geometric averaging over the common marginals to enforce the constraint. We specialize this algorithm to update rules with Gaussian observation models and density estimates. The Gaussian relative position observations are simulated and accuracy is compared to Belief propagation and full state consensus algorithms in varying graph topology.**

I. Introduction

Advances in sensor technology, data processing and communication systems have enabled the deployment of multiple sensors onboard autonomous robots, connected in a communication network [12], [26]. Distributed localization of the sensors in a common reference frame based on relative measurements is a critical capability needed in such networks. Fast and accurate localization forms the basis of services such as target tracking [1], [24], distributed mapping [14], and task assignment [34]. Localization of the robot team by a central entity is prohibitive in large networks due to communication and processing costs, and is fault-susceptible even in small ones. This motivates the design of distributed localization algorithms that rely on local measurements, storage, computation, communication to estimate the sensor states in a global frame.

Related work: Distributed network estimation algorithms include diffusion-based approaches [6], where agents share both observations and estimates, and distributed Bayesian approaches [15], where agents only share densities over the state estimates. Bayesian techniques such as the sum-product algorithm (belief propagation) [33], [31] and consensus pooling [17], [16] can be decomposed into consensus and likelihood update steps. Consensus algorithm estimates converge to true pdf when each agent's prior is (geometrically) averaged [17], [13] with one-hop neighbors before applying the usual Bayesian measurement update. Gaussian distributions versions of the consensus [7], [1] and belief propagation [9], [4] algorithms have been designed and also shown to converge. Distributed Bayesian and consensus pooling algorithms require maintaining and exchanging estimates over entire network's state, which is not feasible in large networks. On the other hand, the belief propagation updates agents' own state estimates based on their edge likelihoods and neighbor marginals. When a factor graph contains cycles, usually present in localization problems, belief propagation does not guarantee convergence. For reducing the stored and communicated data across the network, our previous work [19] details likelihood averaging algorithm for source localization in sensor-specific subsets in discrete setting.

Network localization may be performed with sensors such as radar, infrared, acoustic and cameras [2], [11], [20], modeled via range and bearing models. Several parametric and non-parametric algorithms have been applied to location tracking in range [28], [21] and bearing models [32]. The cooperative approaches require agents to share all estimated sensor positions in the network. In localization problems using relative measurements, the notion of network identifiability determines the existence of a unique solution given the choice of agent observation models. With relative position measurements, the estimated positions are translation invariant, which necessitates at least one anchor node with known position to remove the translation ambiguity [1]. Localization with range-only measurements can be solved up to an isometry [8], [29], while bearing-only localization requires a rigid graph for a unique solution [30]. In this work, we assume that network identifiability holds for the available observation models.

Contributions: This work proposes novel likelihood averaging rules for distributed estimation that enable learning agent densities defined over local neighbor states, exploiting the structure in localization problems with relative (edge) measurements. We derive the proposed algorithm using distributed stochastic mirror descent over the space of pdfs subject to a constraint that the agent pdfs agree over the intersections of their local neighborhoods. The proposed algorithm significantly reduces the storage and communication costs compared to consensus-pooling and distributed Bayesian algorithms. We specialize the general formulation to Gaussian estimates and observation models and show that the algorithm converges in sparse graphs with relative position measurements where Belief propagation fails.

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181, NSF NRI CNS-1830399, FA9550-18-1-0158, and ONR SAI N00014-19-1-2471.

The authors are with Contextual Robotics Institute, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093 pparitos,natanasov,soniamd@ucsd.edu

II. Problem Formulation

We consider a cooperative localization problem over a sensor network with unknown states, where each agent/node receives relative observations from its neighbors and estimates its state in a global reference frame [3]. Denote the state vectors of the n agents as $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, $i \in \{1, \ldots, n\}$, where $\mathcal{X}$ is bounded. The agents communicate via an undirected and connected graph $\mathcal{G}$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$. The neighbors of agent i, including itself, are denoted $\mathcal{V}_i$. The common neighbors of agents i and j are $\mathcal{V}_{ij} \triangleq \mathcal{V}_i \cap \mathcal{V}_j$. We represent the communication graph via an adjacency matrix A with entries $A_{ij} = 0$, if agents i and j are not connected, and $A_{ij} > 0$ for existing edges and self-loops. Every such matrix A representing a connected network can be made doubly stochastic [27].

Each agent's relative measurements with respect to its neighbors at each time t are represented as $\boldsymbol{z}_{ij,t} \in \mathbb{R}^\ell$ and assumed to be a sample of the observation model $\mathrm{q}_i(\boldsymbol{z}_{ij,t}|\boldsymbol{x}_i^*, \boldsymbol{x}_j^*) \in \mathcal{F}_\ell$. The pdf space $\mathcal{F}_\ell$ over a variable of ℓ-dimensions is described as,

$$\mathcal{F}_\ell = \left\{ g \in L^2(\mathbb{R}^\ell) \text{ s.t.} \int_{\mathcal{X}} g(\boldsymbol{x}) d\boldsymbol{x} = 1, g(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^\ell \right\} \quad (1)$$

The vector of all such observations $\boldsymbol{z}_{ij,t}$ at agent i is given as variable $\boldsymbol{z}_{i,t}$, and variable $\boldsymbol{z}_{1:n,t}$ represents the collection of $\boldsymbol{z}_{i,t}$ over all n agents. Similarly, the vector $\boldsymbol{x}_{1:n} \in \mathbb{R}^{nd}$ concatenates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Combining pairwise observations at an agent, the observation model for agent i is $\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i) = \prod_{j \in \mathcal{V}_i} \mathrm{q}_i(\boldsymbol{z}_{ij,t}|\boldsymbol{x}_i, \boldsymbol{x}_j)$. For agent i, the vector $\boldsymbol{y}_i \in \mathbb{R}^{n_i d}$ includes neighbor node states $\{\boldsymbol{x}_i, \{\boldsymbol{x}_j\}_{j \in \mathcal{V}_i}\}$. The observation model for all agents is $\mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n}) \in \mathcal{F}_{n\ell}$ is written as a product in observation space $\prod_{i \in \mathcal{V}} \mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i)$. These observation models may represent range, bearing, position and any other combinations. The true data generating model are defined with true locations $\boldsymbol{y}_i^*, \boldsymbol{x}_{1:n}^*$.

We make the following **assumptions** to further define our problem: (a) The communication network is connected and the graph adjacency matrix A satisfies $A\mathbf{1} = \mathbf{1}, A = A^\top$, and diagonal entries $A_{ii} > 0, \forall i \in \{1, \ldots, n\}$, where $\mathbf{1}$ is a vector of ones. (b) Each agent i estimates a pdf $p_i(\boldsymbol{y}_i)$ over the self and neighbor state variables. (c) Relative measurements are conditionally independent and conditioned on the neighbor nodes following the observation model $\mathrm{q}_i(\boldsymbol{z}_{i,t}|\{\boldsymbol{x}_j\}_{j \in \mathcal{V}_i})$. (d) Relative measurements $\{\boldsymbol{z}_{i,t}\}$ at agent i are independent across time.

Problem 1: The network of agents aims at collectively learning the pdf $\bar{p}(\boldsymbol{x}_{1:n}) \in \mathcal{F}_{nd}$ over the unknown agent positions $\boldsymbol{x}_{1:n}$ using edge measurements $\boldsymbol{z}_{1:n,t}$ collected at each time t.

$$\min_{\bar{p}} \left\{ \mathop{\mathbb{E}}_{(\boldsymbol{x}_{1:n}) \sim \bar{p}} [f(\boldsymbol{x}_{1:n})] \right\}, \quad (2)$$

where $f(\boldsymbol{x}_{1:n}) := \mathrm{D}_\Psi(\mathrm{q}(\cdot|\boldsymbol{x}_{1:n}^*) || \, \mathrm{q}(\cdot|\boldsymbol{x}_{1:n}))$. For density functions $\mathrm{q}(\cdot|\boldsymbol{x}_{1:n}^*), \mathrm{q}(\cdot|\boldsymbol{x}_{1:n}) \in \mathcal{F}_{n\ell}$, the KL-divergence is defined as $\mathrm{D}_\Psi(\mathrm{q}(\cdot|\boldsymbol{x}_{1:n}^*) || \, \mathrm{q}(\cdot|\boldsymbol{x}_{1:n})) = \int_{\mathbb{R}^{n\ell}} \mathrm{q}(\cdot|\boldsymbol{x}_{1:n}^*) \log(\mathrm{q}(\cdot|\boldsymbol{x}_{1:n}^*) / \, \mathrm{q}(\cdot|\boldsymbol{x}_{1:n}))$.

The optimization presented here follows stochastic programming [25] with the caveat that the pdf $\mathrm{q}(\boldsymbol{z}_{1:n,t}) \triangleq \mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n}^*)$ is unknown, but sampled at each time step. We can express the minimization problem defined for finding $\bar{p}(\boldsymbol{x}_{1:n}) \in \mathcal{F}_{nd}$ in terms of sampling averages. We introduce inner product notation $\langle p_1, p_2 \rangle = \int_\Omega p_1 p_2 d\omega$, for $p_1, p_2 \in \mathcal{F}_{nd}$.

$$\begin{aligned}
&\min_{\bar{p} \in \mathcal{F}_{nd}} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n} \sim \bar{p}} [\mathrm{D}_\Psi(\mathrm{q}(\boldsymbol{z}_{1:n,t}) || \, \mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n}))] \right\} \\
&= \min_{\bar{p} \in \mathcal{F}_{nd}} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n} \sim \bar{p}} [-\langle \mathrm{q}(\boldsymbol{z}_{1:n,t}), \log(\mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n})) \rangle] \right\} \quad (3) \\
&= \min_{\bar{p} \in \mathcal{F}_{nd}} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{z}_{1:n,t} \sim \mathrm{q}(\boldsymbol{z}_{1:n,t})} \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n} \sim \bar{p}} [-\log(\mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n}))] \right\} \\
&\qquad \text{(Fubini-Tonelli theorem)} \\
&= \min_{\bar{p}} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{z}_{1:n,t} \sim \mathrm{q}(\boldsymbol{z}_{1:n,t})} F[\bar{p}, \boldsymbol{z}_{1:n,t}] \right\}, \\
&F[\bar{p}; \boldsymbol{z}_{1:n,t}] = \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n} \sim \bar{p}} [-\log(\mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n}))]. \quad (4)
\end{aligned}$$

where the first equality after (3), the entropy term $\int \mathrm{q} \log(\mathrm{q})$ is independent of $\boldsymbol{x}_{1:n}$. The expectation function in (4) can be approximated in terms of sampled data $\boldsymbol{z}_{1:n,t}$ in objective $F[\bar{p}; \boldsymbol{z}_{1:n,t}]$ at different time steps as,

$$\min_{\bar{p} \in \mathcal{F}} \left\{ \frac{1}{T} \sum_{t=1}^{T} F[\bar{p}; \boldsymbol{z}_{1:n,t}] \right\}. \quad (5)$$

III. Convex functionals and derivatives

This section introduces the stochastic mirror descent (SMD) algorithm and its application in function spaces.

A. A primer on the Stochastic Mirror Descent algorithm

The SMD algorithm [5], [18] generalizes stochastic gradient descent (SGD) for convex optimization problems via a divergence operator. Let $g(\boldsymbol{w}; \boldsymbol{v})$, with $\boldsymbol{w} \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^m$, be a real-valued function which is convex in its first argument. Consider the stochastic optimization problem:

$$\min_{\boldsymbol{w}} \mathbb{E}[g(\boldsymbol{w}; \boldsymbol{v})] \approx \frac{1}{T} \sum_{t=1}^{T} g(\boldsymbol{w}; \boldsymbol{v}_t), \quad (6)$$

where $\{\boldsymbol{v}_t\}$ is a random sequence available online. As $T \to \infty$, computating the gradient of the objective function may be infeasible. Instead, the SMD algorithm employs the gradient $\nabla g(\boldsymbol{w}; \boldsymbol{v}_t)$ at time t to perform iterative optimization:

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}} \left\{ \langle \nabla g(\boldsymbol{w}_t, \boldsymbol{v}_t), \boldsymbol{w} \rangle + \frac{1}{\alpha_t} D_\phi(\boldsymbol{w}, \boldsymbol{w}_t) \right\}, \quad (7)$$

where $\langle \cdot, \cdot \rangle$ is the inner product on $\mathbb{R}^n$ and $D_\phi(\boldsymbol{w}, \boldsymbol{w}_t)$ is a *Bregman divergence* between $\boldsymbol{w}$ and $\boldsymbol{w}_t$.

Definition 1: Consider a continuously differentiable and strictly convex function $\phi : \mathcal{X} \to \mathbb{R}$ defined on a convex set $\mathcal{X}$. The *Bregman divergence* associated with ϕ for points $\boldsymbol{w}, \bar{\boldsymbol{w}} \in \mathcal{X}$ is $D_\phi(\boldsymbol{w}, \bar{\boldsymbol{w}}) := \phi(\boldsymbol{w}) - \phi(\bar{\boldsymbol{w}}) - \langle \nabla \phi(\bar{\boldsymbol{w}}), \boldsymbol{w} - \bar{\boldsymbol{w}} \rangle$.

Choosing $\phi(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2$, makes D_ϕ the squared Euclidean distance and (7) becomes the standard SGD algorithm. The sequence $\{\alpha_t\}_{t \geq 1}$ should be non-summable but

square-summable to ensure convergence. The convergence rate for minimizing convex functions is $\mathcal{O}(\frac{1}{\sqrt{T}})$, independently of the problem dimension [18].

B. Derivatives and Bregman divergence in functional space

The problem considered in this paper (5) is defined over the function space $\mathcal{F}_{nd}$ of probability density functions $\bar{p}$. Applying SMD to (5) requires generalizing the terms in (7).

Consider functions $g_1, g_2 \in L^2(\mathcal{X}^n)$. The standard inner product on $L^2(\mathcal{X}^n)$ is $\langle g_1, g_2 \rangle := \int g_1 g_2 d\mu$, where μ is the Lebesgue measure on $\mathcal{X}^n$. A subset $\mathcal{A}$ of $L^2(\mathcal{X}^n)$ is convex if and only if $\alpha g_1 + (1-\alpha) g_2 \in \mathcal{A}$ for any $g_1, g_2 \in \mathcal{A}$ and $\alpha \in [0,1]$. The set of pdfs $\mathcal{F}_{nd}$ in (1) is a closed convex subset of $L^2(\mathcal{X}^n)$. To define a divergence operator over $\mathcal{F}_{nd}$, we consider the entropy functional $\Psi[g] = \int g \log(g)$ for $g \in \mathcal{F}_{nd}$. Entropy is continuously differentiable and strictly convex as (i) $\mathcal{F}_{nd}$ is convex and (ii) $x \log(x)$ in x is strictly convex over the positive real numbers, and the integration operator is linear, so it holds that $\Psi[\alpha g_1 + (1-\alpha) g_2] < \alpha \Psi[g_1] + (1-\alpha)\Psi[g_2]$ for all $g_1, g_2 \in \mathcal{F}_{nd}$, $g_1 \neq g_2$ a.e. The Bregman divergence associated with Ψ is the *Kullback-Leibler divergence* $D_\Psi(g_1, g_2) := \int g_1 \log(g_1/g_2)$ [10].

To finish the extension of SMD to $\mathcal{F}_{nd}$, we need a definition of functional derivative. To evaluate how a functional F changes in the vicinity of $g \in L^2(\mathcal{X}^n)$, we consider variations of g defined as $g + \epsilon\eta$, where $\eta \in L^2(\mathcal{X}^n)$ and $\epsilon \geq 0$ is a small scalar. For fixed g, η, $F[g+\epsilon\eta]$ is a function of ϵ and limits can be evaluated in the usual sense.

Definition 2: A functional $F : L^2(\mathcal{X}^n) \to \mathbb{R}$ is *Gateaux differentiable* at $g \in L^2(\mathcal{X}^n)$, if

$$F'[g;\eta] := \lim_{\epsilon \to 0^+} \frac{F[g+\epsilon\eta] - F[g]}{\epsilon}. \tag{8}$$

exists for any $\eta \in L^2(\mathcal{X}^n)$ and there is an element $\frac{\delta F}{\delta g} \in L^2(\mathcal{X}^n)$ such that $\int \frac{\delta F}{\delta g} \eta = F'[g;\eta]$.

Proposition 1: For $p, g \in \mathcal{F}_{nd}$, the Gateaux derivative of:

1) $L[p] = \langle p, g \rangle$ is $\frac{\delta L}{\delta p} = g$,
2) $\Psi[p] = \langle p, \log(p) \rangle$ is $\frac{\delta \Psi}{\delta p} = 1 + \log p$.
3) $\mathrm{D}_\Psi(p||g) = \langle p, \log(p/g) \rangle$ is $\frac{\delta \mathrm{D}_\Psi}{\delta p} = 1 + \log(p/g)$.

Proof: The derivative of $L[p]$ follows by the definition. The directional derivative of $\Psi[p]$ along $\eta \in \mathcal{F}$ is,

$$\Psi'[g;\eta] = \lim_{\epsilon \to 0^+} \frac{1}{\epsilon} \left(\int (g+\epsilon\eta)\log(g+\epsilon\eta) - \int g \log g \right)$$
$$= \int \lim_{\epsilon \to 0^+} \frac{1}{\epsilon} ((g+\epsilon\eta)\log(g+\epsilon\eta) - g\log g) = \int \eta \log(g) + \eta,$$

where we use the dominated convergence theorem [23] to exchange the limit with the integral (as ϵ can be taken to be $0 \leq \epsilon \leq 1$, we have that $(f+\epsilon\eta)\log(f+\epsilon\eta) \leq (f+\eta)\log(f+\eta)$, which is an integrable function). The derivations for linear and entropy functionals may be summed to obtain the KL-divergence derivative. ∎

IV. Stochastic Mirror Descent over the Probability Simplex

This section derives the existing distributed algorithms for extensions to marginal space updates. We apply the SMD algorithm to an optimization problem over the probability simplex $\mathcal{F}$ in the form specified in (3). We start with deriving the additive structure of $F[p; \boldsymbol{z}_{1:n,t}] = \sum_{i=1}^n F_i[p; \boldsymbol{z}_{i,t}]$, where $F_i[p; \boldsymbol{z}] := -\langle \log \mathrm{q}_i(\boldsymbol{z}|\cdot), p(\cdot) \rangle$ and exploit this to obtain a distributed SMD formulation.

A. Centralized SMD

With $\{p_t\}_{t \geq 0}$ as iterates of SMD algorithm and $\mathrm{D}_\Psi(p||p_t)$ as the Bregman divergence between $p, p_t \in \mathcal{F}_{nd}$ (c.f. Sec. III), the SMD algorithm applied to Eqn. (5) is:

$$p_{t+1} = \arg\min_{p \in \mathcal{F}} \left\{ \left\langle \frac{\delta F}{\delta p}[p_t, \boldsymbol{z}_{1:n,t}], p \right\rangle + \frac{1}{\alpha_t} \mathrm{D}_\Psi(p||p_t) \right\}. \tag{9}$$

Proposition 2: The optimization problem in (9) has a closed-form solution:

$$p_{t+1}(\cdot) \propto \exp\left(-\alpha_t \frac{\delta F}{\delta p}[p_t(\cdot); \boldsymbol{z}_{1:n,t}] \right) p_t(\cdot). \tag{10}$$

Proof: Problem (9) is an equality-constrained optimization over the pdf space $\mathcal{F}_{nd}$. To take the constraint $\int p = \langle 1, p \rangle = 1$ into account, we consider the Lagrangian:

$$\mathcal{L}(p, \lambda) = \left\langle \frac{\delta F}{\delta p}[p_t, \boldsymbol{z}_{1:n,t}], p \right\rangle + \frac{1}{\alpha_t} \mathrm{D}_\Psi(p||p_t) + \lambda \left(\langle 1, p \rangle - 1 \right),$$

where λ is a multiplier. The variation of $\mathcal{L}$ w.r.t. p is:

$$\frac{\delta \mathcal{L}}{\delta p} = \frac{\delta F}{\delta p}[p_t, \boldsymbol{z}_{1:n,t}] + \frac{1}{\alpha_t} (1 + \log p - \log p_t) + \lambda.$$

Setting the variation to zero and solving for p leads to:

$$p(\cdot) = e^{-1-\alpha_t\lambda - \alpha_t \frac{\delta F}{\delta p}[p_t(\cdot), \boldsymbol{z}_{1:n,t}]} p_t(\cdot).$$

The value of λ can be obtained from the constraint:

$$1 = \int p(\cdot) = e^{-1-\alpha_t\lambda} \underbrace{\int e^{-\alpha_t \frac{\delta F}{\delta p}(p_t(\cdot), \boldsymbol{z}_{1:n,t})} p_t(\cdot)}_{Z},$$

showing that $p(\cdot) = \frac{1}{Z} \exp\left(-\alpha_t \frac{\delta F}{\delta p}(p_t(\cdot), \boldsymbol{z}_{1:n,t}) \right) p_t(\cdot)$. ∎

For our specific choice of $F[p, \boldsymbol{z}_{1:n,t}]$,

$$\frac{\delta F}{\delta p}[p_t(\cdot), \boldsymbol{z}_{1:n,t}] = -\frac{\delta}{\delta p} \langle \log \mathrm{q}(\boldsymbol{z}_{1:n,t}|\cdot), p(\cdot) \rangle \Big|_{p(\cdot)=p_t(\cdot)}$$
$$= -\log \mathrm{q}(\boldsymbol{z}_{1:n,t}|\ \cdot). \tag{11}$$

Applying Prop. 2 leads to the following SMD algorithm over the probability simplex:

$$p_{t+1}(\boldsymbol{x}_{1:n}) \propto \mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{x}_{1:n})^{\alpha_t} p_t(\boldsymbol{x}_{1:n}). \tag{12}$$

B. Distributed SMD

Instead of solving the optimization problem (5) centrally, we observe the additive structure of F to obtain a distributed formulation, in which agents keep local estimates $p_{i,t} \in \mathcal{F}_{nd}$ of the state pdf. The key idea was proposed in [16] based on the observation that we can introduce new agent-specific variables $\bar{p}_i = \bar{p}$ in (2), along with a constraint that their values agree, $\bar{p}_i = \bar{p}_j$ for all $i, j \in \mathcal{V}$. Using the nodal independence of $\boldsymbol{z}_i$ in observation models $\log(\mathrm{q}(\boldsymbol{z}_{1:n}|\boldsymbol{x}_{1:n})) = \sum_{i=1}^n \log(\mathrm{q}_i(\boldsymbol{z}_i|\boldsymbol{x}_{1:n}))$ and data generating densities $\mathrm{q}(\boldsymbol{z}_{1:n,t}) = \prod_{i \in \mathcal{V}} \mathrm{q}_i(\boldsymbol{z}_{i,t})$

on Eqn. 4, the distributed objective satisfies $F[\bar{p}, \boldsymbol{z}_{1:n}] = \sum_{i=1}^{n} F_i[\bar{p}_i, \boldsymbol{z}_{i,t}]$ is,

$$F_i[\bar{p}_i, \boldsymbol{z}_{i,t}] = \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim\bar{p}_i}[-\log(\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{x}_{1:n}))]. \quad (13)$$

The agent-wise optimization yields the same results as centralized upon enforcing the constraint $\bar{p}_i = \bar{p}_j, \forall i, j \in \mathcal{V}$.

The SMD algorithm can now be applied locally at each agent i with iterates $p_{i,t}(\boldsymbol{x}_{1:n})$. To eventually enforce the agreement constraint, $\bar{p}_i = \bar{p}_j$, [16] uses geometric averaging with stochastic weights A_{ij}, depending on the network structure, of the pdfs $p_{i,t}$:

$$p_{i,t}(\cdot) \propto \prod_{j\in\mathcal{V}_i} (p_{j,t}(\cdot))^{A_{ij}}.$$

Thus, the local SMD problem becomes:

$$\min_{p\in\mathcal{F}_{nd}} \left\{ -\langle \log \mathrm{q}_i(\boldsymbol{z}_{i,t}|\cdot), p\rangle + \frac{1}{\alpha_{i,t}} D_{\mathrm{KL}}\big(p, \prod_{j\in\mathcal{V}_i} p_{j,t}^{A_{ij}}\big) \right\}, \quad (14)$$

leading to the distributed update rule:

$$p_{i,t+1}(\boldsymbol{x}_{1:n}) \propto \mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{x}_{1:n})^{\alpha_{i,t}} \prod_{j\in\mathcal{V}_i} p_{j,t}(\boldsymbol{x}_{1:n})^{A_{ij}}. \quad (15)$$

The convergence of the algorithm in (15) has been shown in [16]. Applying (15) to the localization problem, leads to a distributed algorithm but the amount of information maintained and exchanged by the agents is still nd dimensional because $p_{i,t} \in \mathcal{F}_{nd}$. This significant memory and communication requirement is counter-intuitive because agents are interested in estimating only their own states $\boldsymbol{x}_i \in \mathbb{R}^d$ and do not necessarily care about the states of the *entire* network. Our main contribution in the following section allows pdfs defined only over the local neighborhood of i to be maintained and exchanged.

V. Proposed marginal consensus estimation algorithm in partial spaces

For localization networks with relative measurements, the pdfs are defined on the partial space of local neighborhood states $\boldsymbol{y}_i$. This section derives the objective function based on observation models in $\boldsymbol{y}_i$, propose an algorithm and specialize it for Gaussian densities. We write the centralized objective in terms of observation models $\mathrm{p}_i(\cdot|\boldsymbol{y}_i)$, and follow the steps in distributed SMD section to decompose the function along agent i's observations $\boldsymbol{z}_{i,t}$. We start with showing that the centralized estimation of node states in equations (2) to (3) can be distributed along agents in partial neighbor state space given by $\boldsymbol{y}_i$.

$$\begin{aligned}&\min_{\bar{p}} \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim\bar{p}}[\mathrm{D}_\Psi(\mathrm{q}(\boldsymbol{z}_{1:n,t})||\,\mathrm{q}(\boldsymbol{z}_{1:n,t}|\boldsymbol{y}_i))]\\ &= \sum_{i\in\mathcal{V}} \min_{\bar{p}_i} \mathop{\mathbb{E}}_{\boldsymbol{y}_i\sim\bar{p}_i}[\mathrm{D}_\Psi(\mathrm{q}_i(\boldsymbol{z}_{i,t})||\,\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i)].\end{aligned}$$

The decomposition leads to a **distributed version** of (3) by introducing pdfs' $p_i(\boldsymbol{y}_i)$ along with constraints ensuring agreement on common states estimated by any two agents.

$$\min_{\bar{p}} F[\bar{p}; \boldsymbol{z}_t] = \sum_{i\in\mathcal{V}} \min_{\bar{p}_i} \mathop{\mathbb{E}}_{\boldsymbol{y}_i\sim\bar{p}_i}[-\langle \mathrm{q}_i(\boldsymbol{z}_{i,t}), \log \mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i)\rangle],$$

further leading to the approximation,

$$\begin{aligned}\min_{\bar{p}} F[\bar{p}; \boldsymbol{z}_t] &= \sum_{i\in\mathcal{V}} \min_{\bar{p}_i} \mathop{\mathbb{E}}_{\boldsymbol{y}_i\sim\bar{p}_i} \mathop{\mathbb{E}}_{\boldsymbol{z}_{i,t}\sim q_i}[-\log(\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i))]\\ &= \sum_{i\in\mathcal{V}} \min_{\bar{p}_i} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{y}_i\sim\bar{p}_i}[-\log(\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i))].\end{aligned}$$

The distributed objective function then becomes,

$$F_i[\bar{p}_i; \boldsymbol{z}_{i,t}] = \mathop{\mathbb{E}}_{\boldsymbol{y}_i\sim\bar{p}_i}[-\log(\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i))]. \quad (16)$$

The SMD algorithm locally learns pdf over local neighbor variables $p_{i,t}(\boldsymbol{y}_i)$ with likelihood update as the Gateaux derivative $\frac{\delta}{\delta\bar{p}_i} F_i[\bar{p}_i; \boldsymbol{z}_{i,t}] = -\log(\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i))$, the same as presented in (15). The agent wise optimization matches the centralized estimates under the constraints $\bar{p}_i = \int_{\mathcal{X}\setminus\mathcal{X}_i} \bar{p}$, which depend on the unknown global $\bar{p}$. Therefore, the constraints are instead represented as agreement in marginal agent pdfs $\bar{p}_i, \forall i \in \mathcal{V}$ over a common space,

$$\int \bar{p}_i d\boldsymbol{x}_k|_{k\in\mathcal{V}_i\setminus\mathcal{V}_{ij}} = \int \bar{p}_j d\boldsymbol{x}_k|_{k\in\mathcal{V}_j\setminus\mathcal{V}_{ij}},\ \forall (i,j) \in \mathcal{E}.$$

Since every agent's state is estimated by its one and two-hop neighbors, all marginals of the estimated pdfs on any common set of variables are equal. These constraints requiring marginal agreement on edges are enforced with geometric averaging on self-conditional and neighbor-marginals product $\tilde{p}_{ji,t}$ as detailed below. Each agent messages marginal density estimates $p_{ji,t}$ over common states to neighbors.

$$\begin{aligned}p_{i,t} &\propto \prod_{j\in\mathcal{V}_i} (\tilde{p}_{ji,t})^{A_{ij}},\\ \tilde{p}_{ji,t} &= p_{i,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_i\setminus\mathcal{V}_{ij}}|\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}}) p_{ji,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}}),\\ p_{ji,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}}) &= \int p_{j,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_j}) d\boldsymbol{x}_{k|k\in\mathcal{V}_j\setminus\mathcal{V}_{ij}}.\end{aligned}$$

Our Algorithm 1 thus consists of the following steps: **edge merging**, **geometric pooling**, **likelihood update** and **message generation** performed by each agent at each time step.

Data: estimate $p_{i,t}(\boldsymbol{y}_i)$, weights $\{A_{ij}\}_{j\in\mathcal{V}_i}$, neighbor messages $p_{ji,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}})$, measurement $\boldsymbol{z}_{i,t}$, measurement model $\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i)$

`// Combine neighbor estimates.`

for $j \in \mathcal{V}_i$ **do**

  Product of j's marginal and i's conditional: $\tilde{p}_{ji,t} = p_{i,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_i\setminus\mathcal{V}_{ij}}|\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}}) p_{ji,t}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}})$;

Weighted average: $\tilde{p}_{i,t}(\boldsymbol{y}_i) := \prod_{j\in\mathcal{V}_i} \tilde{p}_{ji,t}(\boldsymbol{y}_i)^{A_{ij}}$;

`// Bayesian update.`

$p_{i,t+1}(\boldsymbol{y}_i) = \mathrm{q}_i(\boldsymbol{z}_{i,t+1}|\boldsymbol{y}_i)\tilde{p}_{i,t}(\boldsymbol{y}_i)$;

`// Generate neighbor messages.`

for $j \in \mathcal{V}_i$ **do**

  Find common neighbor marginal $p_{ij,t+1}(\{\boldsymbol{x}_k\}_{k\in\mathcal{V}_{ij}}) = \int_{\mathcal{V}\setminus\mathcal{V}_{ij}} p_{i,t+1}(\boldsymbol{y}_i)$;

Algorithm 1: Marginal density averaging at agent i

In comparison to distributed algorithm in Section IV-B, this algorithm reduces the size of the communicated messages from a density over all the network states in $\mathcal{V}$ to a partial common set $\mathcal{V}_{ij}$ between sensors (i, j). Whereas, on computational front, there is an added step for finding the conditional density at each node. This trade-off depends on the number of connections in the sensor network.

Gaussian marginal density averaging

This section specializes our proposed Algorithm 1 to Gaussian marginal updates. We present the Gaussian estimate equivalent to the four algorithm steps in the following lemmas. Here, we denote a Gaussian random variable with mean μ and information matrix as Ω $\mathcal{N}(\mu, \Omega^{-1})$, and its associated density function as $\phi(\cdot|\mu, \Omega^{-1})$.

Lemma 1 (Neighbor messages): The marginal density of the Gaussian pdf $\phi\left(\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}, \begin{bmatrix}\Omega_{11} & \Omega_{12}\\ \Omega_{21} & \Omega_{22}\end{bmatrix}^{-1}\right)$ with respect to $\boldsymbol{x}_1$ is given as,

$$\phi(\boldsymbol{x}_1\Big|\boldsymbol{\mu}_1, \left(\Omega_{11} - \Omega_{12}\Omega_{22}^{-1}\Omega_{21}\right)^{-1}).$$

Lemma 2 (Pre-edge merging): Let (X_1, X_2) be random vectors represented by a joint Gaussian distribution with mean $\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}$ and information matrix $\Omega = \begin{bmatrix}\Omega_{11} & \Omega_{12}\\ \Omega_{21} & \Omega_{22}\end{bmatrix}$. The pdf associated with conditional distribution is,

$$(X_1|X_2 = \boldsymbol{x}_2) \sim \mathcal{N}\left(\mu_1 - \Omega_{11}^{-1}\Omega_{12}(\boldsymbol{x}_2 - \mu_2), \Omega_{11}^{-1}\right).$$

Proposition 3 (Edge merging): Let X_1, X_2 be random vectors with a joint Gaussian distribution. Assume that X_1 conditioned on $X_2 = x_2$ is distributed as $\mathcal{N}(\mu_1 - \Omega_{11}^{-1}\Omega_{12}(x_2 - \mu_2), \Omega_{11}^{-1})$ and that the marginal distribution of X_2 is $\mathcal{N}(\bar{\mu}_2, \bar{\Omega}_{22}^{-1})$. Then, X_1 and X_2 joint distribution is

$$\mathcal{N}\left(\begin{bmatrix}\mu_1 + \Omega_{11}^{-1}\Omega_{12}(\mu_2 - \bar{\mu}_2)\\ \bar{\mu}_2\end{bmatrix}, \begin{bmatrix}\Omega_{11} & \Omega_{12}\\ \Omega_{12}^\top & \bar{\Omega}_{22} + \Omega_{12}^\top\Omega_{11}^{-1}\Omega_{12}\end{bmatrix}^{-1}\right).$$

Lemma 3 (Geometric averaging): Let $\Omega_w = \sum_{i=1}^n A_i\Omega_i$. The weighted geometric product of Gaussian density functions $\phi(\boldsymbol{x}|\mu_i, \Omega_i^{-1}), \forall i \in \{1, \ldots, n\}$ with corresponding weights A_i is given as,

$$\prod_{i=1}^n \phi(\boldsymbol{x}|\mu_i, \Omega_i^{-1})^{A_i} = \phi\left(\boldsymbol{x}\Big|\Omega_w^{-1}\sum_{i=1}^n A_i\Omega_i\mu_i, \Omega_w^{-1}\right).$$

Lemma 4 (Likelihood update): Let the likelihood density be described as $\mathrm{q}_i(\boldsymbol{z}_{i,t}|\boldsymbol{y}_i) = \phi\left(\boldsymbol{z}_{i,t}|H_i\boldsymbol{y}_i, V_i\right)$. Then the posterior Gaussian density obtained as likelihood prior product $\phi\left(\boldsymbol{z}_{i,t}|H_i\boldsymbol{y}_i, V_i^{-1}\right)\phi\left(\boldsymbol{y}_i; \mu, \Omega_i^{-1}\right)$ is

$$\mathcal{N}\left(\left(H_i^\top V_i H_i + \Omega_i\right)^{-1}(H_i^\top V_i\boldsymbol{z}_{i,t} + \Omega_i\mu_i),\ \left(H_i^\top V_i H_i + \Omega_i\right)^{-1}\right)$$

These results lead to the closed-form implementation as described in Algorithm 1.

VI. SIMULATIONS

A. Cooperative localization with relative measurements

Consider a network of 10 nodes with unknown locations $\boldsymbol{x} = [\boldsymbol{x}_i]_{i\in\mathcal{V}}, \boldsymbol{x}_i \in \mathbb{R}^2$, which perform cooperative localization via noisy relative measurements. To guarantee network identifiability, and since the states lie in a two-dimensional space, we assume there is one unique anchor node with a known position in the network. Each node i obtains relative measurements from node j of the form $\boldsymbol{z}_{ij} = (\boldsymbol{x}_j - \boldsymbol{x}_i) + \epsilon, \epsilon \sim \mathcal{N}(0, V_i), V_i = \mathbb{I}_d$. The collected measurements at node i at time t are given as $\boldsymbol{z}_{i,t} = [\boldsymbol{z}_{ij,t}]_{j\in\mathcal{V}_i}$.

1) Full state estimation (FS): The Gaussian form of the distributed SMD for the state vector $\boldsymbol{x} = [\boldsymbol{x}_1^\top \ldots \boldsymbol{x}_n^\top]^\top$ and observation model $\mathrm{q}_i(\boldsymbol{x}) = \mathcal{N}(H_i\boldsymbol{x}, V_i)$ for each $i \in \mathcal{V}$, is given in [1]. The update rules employ the mean $\mu_{i,t}$ and information matrix $\Omega_{i,t}(= \Sigma_{i,t}^{-1})$ for each agent as

$$\Omega_{i,t} = \sum_{j\in\mathcal{V}_i}\Omega_{j,t-1};\ \mu_{i,t} = \Omega_{i,t}^{-1}(\sum_{j\in\mathcal{V}_i}\Omega_{j,t-1}\mu_{j,t-1}).$$

2) Belief propagation (BP): This long-established, distributed algorithm aims to compute a joint probability distribution of the form $\prod_{i=1}^n p_i(\boldsymbol{x}_i)$, where $p_i(\boldsymbol{x}_i)$ (with $\sum_{i=1}^n p_i(\boldsymbol{x}) = 1$) refers to the pdf representing the agent estimate of $\boldsymbol{x}_i$ in the network. In absence of any loops in the communication network, BP [22] has been shown to estimate the correct marginals at each node. BP implements message generation and pooling steps as follows

$$m_{t,ij}(\boldsymbol{x}_i) = \sum_{x_i}\mathrm{q}_i(\boldsymbol{z}_{ij}|\boldsymbol{x}_i, \boldsymbol{x}_j)p_{i,t}(\boldsymbol{x}_i)\prod_{k\in\mathcal{V}_j\setminus i}m_{t-1,kj}(\boldsymbol{x}_i),$$

$$p_{i,t}(\boldsymbol{x}_i) = \frac{p_{i,t-1}(\boldsymbol{x}_i)\prod_{k\in\mathcal{V}_i}m_{ki}(\boldsymbol{x}_i)}{\sum_{j=1}^n p_{j,t-1}(\boldsymbol{x}_j)\prod_{k\in\mathcal{V}_j}m_{kj}(\boldsymbol{x}_j)}.$$

BP incurs into a significantly larger number of messages passed between agents as compared to the method presented here.

A Gaussian BP algorithm is described in [4] for agents with observation model $\boldsymbol{z}_i = H\begin{bmatrix}\boldsymbol{x}_i & \boldsymbol{x}_j\end{bmatrix}^\top + \epsilon, \epsilon \sim \mathcal{N}(0_{d\times 1}, \Omega_i^z)$, with $H = \begin{bmatrix}-1, 1\end{bmatrix}\otimes\mathbb{I}_d$, where $\otimes$ is a kronecker product. The update rule for each agent is given as

$$\Omega_{jj,t} = \sum_{i\in\mathcal{V}_j}\Omega_{ij,t-1};\ \mu_{jj,t} = \Omega_{jj,t}^{-1}\left(\sum_{i\in\mathcal{V}_j}\Omega_{ij,t-1}\mu_{ij,t-1}\right),$$

which depends on the messages sent to j from $i \in \mathcal{V}_j$:

$$\Omega_{ij,t} = \begin{bmatrix}\Omega_{ii,t} - \Omega_{ji,t-1} & 0\\ 0 & 0\end{bmatrix} + H_i^\top\Omega_i^z H_i,$$

$$\mu_{ij,t} = \Omega_{ij,t}^{-1}\left(\begin{bmatrix}\sum_{k\in\{\mathcal{V}_i\setminus j\}}\Omega_{ki,t-1}\mu_{ki,t-1}\\ 0\end{bmatrix} + H_i^\top\Omega_i^z\boldsymbol{z}_{ij,t}\right).$$

3) Algorithm comparison: We compare our algorithm for Gaussian estimates with the previous ones. Figure 1 presents the performance of the three algorithms for a ring network. As expected, the FS estimation performs better than our algorithm based on partial updates, which outperforms BP. In fact BP fails to converge in a reasonable amount of time,

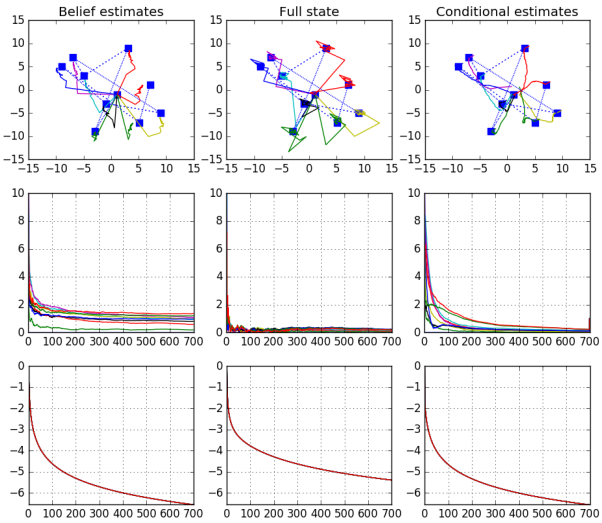

Fig. 1. (Row 1) Convergence of self-estimates to true agent positions. (Row 2) Estimate error for each agent's self estimate across time. (Row 3) Maximum eigenvalue of the self-covariance estimates in BP, full, and partial state estimation algorithms for a ring network of 10 agents.

TABLE I

COMPARING THE ITERATIONS AND COMMUNICATED NUMBERS FOR CONVERGENCE TO A FIXED ERROR IN A 25-NODE GRAPH

	Iterations			Information units		
	BP	FS	CS	BP	FS	CS
Line	NA	18	1356	NA	2203k	1301k
100 edges	9	2	28	29.8k	846.6k	291k
287 edges	7	2	15	51k	1856k	2709k

whereas our algorithm does. Based on the log singular value updates of self-covariance, BP and the proposed algorithm exhibit more confidence in their estimates compared to the full-state algorithm. We next compare the amount of shared information needed to reach the same estimated error level in Table I over a 25-node network. We define that communicating d-dimensional Gaussian densities requires sharing $d+d^2$ units corresponding to the mean and covariance respectively. Even though the proposed algorithm requires more iterations for convergence, fewer information units are shared over the network than full state iterations in relatively sparse graphs.

Figure 2 presents the results for a large-scale network of 100 nodes and 400 edges, showing convergence in each case. For dense graphs, we observe that BP converges faster than our proposed algorithm. We also observe that the log singular value of the covariance matrix converges faster for BP than for other methods. This confidence does not coincide with the quality observed in the full-state updates. The computation complexity of our algorithm is contingent upon the number of matrix inverses. We observe that computational time progresses from BP, full-state estimation and our algorithm.

To further contrast our algorithm and BP, we simulated several 10 node networks with increasing number of edges in $\{9, \ldots, 45\}$. In all cases, we observed that the full-state algorithm outperformed the other two algorithms wrt final error estimates. An interesting pattern was observed between BP and proposed algorithm. The latter outperformed

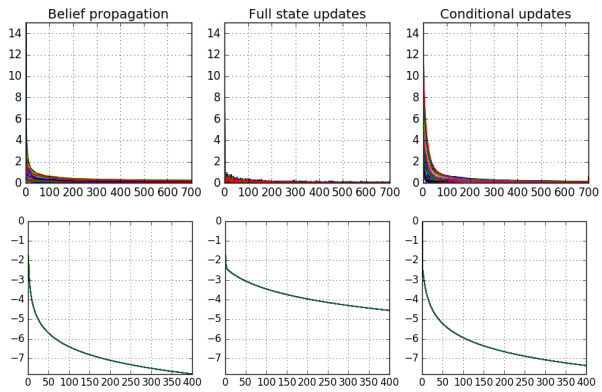

Fig. 2. (Row 1) Self position estimates among agents and (Row 2) Maximum eigenvalue of the self-covariance estimates in a 100-node, 400-edge connected network via BP, full, and partial state estimation algorithms.

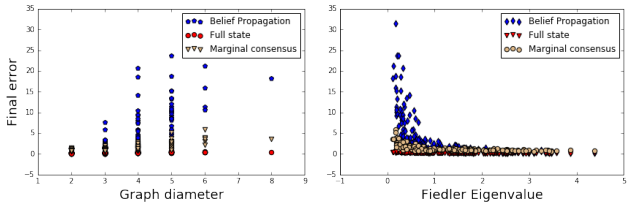

Fig. 3. Error in self position estimates via BP, full and partial state estimation algorithms. (a) With increasing graph diameter after 500 steps. (b) With increasing connectivity captured by Fiedler Eigenvalue.

BP accuracy for lower connectivity networks with fewer edges. Figure 3 shows how lower connectivity measured by Fiedler eigenvalue implies that BP performs worse than our algorithm, while being comparable in denser graphs.

VII. CONCLUSIONS

This paper presents a distributed and scalable estimation algorithm to solve cooperative localization problem. We pose this problem as a joint optimization in space of pdfs, as a pdf formulation allows for characterizing higher order moments of the estimated positions. The optimization is subject to constraints over estimated pdf's marginals to ensure estimation consistency. The proposed algorithm employs a novel edge averaging estimation step that follows likelihood update in SMD optimization. We then specialize this algorithm to Gaussian estimates and compare its performance with well established Belief propagation and full state estimation distributed algorithms. Simulations show that our algorithm converges on sparse networks, where Belief propagation fails, and remains competitive w.r.t. other cases.

REFERENCES

[1] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas. Joint estimation and localization in sensor networks. In *IEEE Conf. on Decision and Control*, pages 6875–6882, 2014.

[2] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and data fusion: a handbook of algorithms*. YBS Publishing, 2011.

[3] P. Barooah and J. P. Hespanha. Estimation on graphs from relative measurements. *IEEE Control Systems Magazine*, 27(4):57–74, 2007.

[4] D. Bickson. Gaussian belief propagation: Theory and aplication. *arXiv preprint arXiv:0811.2518*, 2008.

[5] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[6] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[7] S. Das and J. M. Moura. Consensus + innovations distributed Kalman filter with optimized gains. *IEEE Transactions on Signal Processing*, 65(2):467–481, 2016.

[8] C. Di Franco, A. Prorok, N. Atanasov, B. Kempke, P. Dutta, V. Kumar, and G. J. Pappas. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *ACM/IEEE Int. Conf. on Info. Proc. in Sensor Networks (IPSN)*, pages 235–246, 2017.

[9] J. Du, S. Ma, Y.-C. Wu, S. Kar, and J. M. Moura. Convergence analysis of distributed inference with vector-valued Gaussian belief propagation. *The Journal of Machine Learning Research*, 18(1):6302–6339, 2017.

[10] B. A. Frigyik, S. Srivastava, and M. R. Gupta. Functional Bregman divergence and Bayesian estimation of distributions. *IEEE Transactions on Information Theory*, 54(11):5130–5139, 2008.

[11] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar. Distributed localization of networked cameras. In *Proceedings of the 5th Int. Conf. on Info. Proc. in sensor networks (IPSN)*, pages 34–42, 2006.

[12] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. *Networked Control Systems*, pages 75–107, 2010.

[13] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi. Non-Bayesian social learning. *Games and Economic Behavior*, 76(1):210–225, Sept. 2012.

[14] B. J. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: an information-theoretic approach. *The International Journal of Robotics Research*, 31(10):1134–1154, 2012.

[15] T. Minka. Divergence measures and message passing. Technical report, 2005. MSR-TR-2005-173.

[16] A. Nedić, A. Olshevsky, and C. A. Uribe. Distributed learning for cooperative inference. *arXiv:1704.02718*, 2017.

[17] A. Nedić, A. Olshevsky, and C. A. Uribe. Fast convergence rates for distributed non-bayesian learning. *IEEE Transactions on Automatic Control*, 62(11):5538–5553, 2017.

[18] A. Nemirovsky and D. Yudin. *Problem complexity and method efficiency in optimization.* Wiley, 1983.

[19] P. Paritosh, N. Atanasov, and S. Martínez. Hypothesis assignment and partial likelihood averaging for cooperative estimation. In *IEEE Int. Conf. on Decision and Control*, pages 7850–7856, Nice, France, December 2019.

[20] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.

[21] L. Paull, S. Saeedi, M. Seto, and H. Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2013.

[22] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, pages 133–136, 1982.

[23] W. Rudin. *Real and complex analysis*. McGraw-Hill, 2006.

[24] V. Savic, H. Wymeersch, and E. G. Larsson. Target tracking in confined environments with uncertain sensor positions. *IEEE Transactions on Vehicular Technology*, 65(2):870–882, 2015.

[25] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014.

[26] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya. Location of Things (LoT): A review and taxonomy of sensors localization in IoT infrastructure. *IEEE Comm. Surveys & Tutorials*, 20(3):2028–2061, 2018.

[27] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[28] B. Teague, Z. Liu, F. Meyer, A. Conti, and M. Z. Win. Peregrine: Network localization and navigation with scalable inference and efficient operation. *arXiv preprint arXiv:2001.11494*, 2020.

[29] P. Tecchio, N. Atanasov, S. Shahrampour, and G. J. Pappas. N-dimensional distributed network localization with noisy range measurements and arbitrary anchor placement. In *American Control Conference (ACC)*, pages 1195–1201, 2019.

[30] R. Tron, L. Carlone, F. Dellaert, and K. Daniilidis. Rigid components identification and rigidity control in bearing-only localization using the graph cycle basis. In *American Control Conference (ACC)*, pages 3911–3918, 2015.

[31] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[32] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3–19, 2013.

[33] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7):2282–2312, 2005.

[34] M. M. Zavlanos and G. J. Pappas. Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, 24(1):232–242, 2008.

APPENDIX

PROOF OF PROPOSITION 3

Let X_1, X_2 be random vectors with a joint Gaussian distribution. Assume that X_1 conditioned on $X_2 = x_2$ is distributed as $\mathcal{N}(\mu_1 - \Omega_{11}^{-1}\Omega_{12}(x_2 - \mu_2), \Omega_{11}^{-1})$ and that the marginal distribution of X_2 is $\mathcal{N}(\bar{\mu}_2, \bar{\Omega}_{22}^{-1})$.

Using the reparametrization trick, $X_1|X_2 = x_2$ can be expressed as the sum of a deterministic "mean" component and a zero-mean random variable $Y \sim \mathcal{N}(0, \Omega_{11}^{-1})$:

$$X_1 = \Omega_{11}^{-1}\Omega_{12}\boldsymbol{x}_2 + \mu_1 + \Omega_{11}^{-1}\Omega_{12}\mu_2 + Y$$

Let us now compute the product of the conditional $p_1(\boldsymbol{x}_1|\boldsymbol{x}_2)$ with the marginal $p_2(\boldsymbol{x}_2)$,

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}_2] &= \bar{\mu}_2, \\
\mathbb{E}[\boldsymbol{x}_1] &= \mathbb{E}_{\boldsymbol{x}_2}[\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_2]] \\
&= \int_{\boldsymbol{x}_2} p_2(\boldsymbol{x}_2)\left(\Omega_{11}^{-1}\Omega_{12}(\mu_2 - \boldsymbol{x}_2) + \mu_1\right) d\boldsymbol{x}_2 \\
&= \Omega_{11}^{-1}\Omega_{12}(\mu_2 - \bar{\mu}_2) + \mu_1.
\end{aligned}$$

We can now compute the covariance matrix of the new distribution. We know that $\mathbb{E}[(\boldsymbol{x}_2 - \bar{\mu}_2)(\boldsymbol{x}_2 - \bar{\mu}_2)^\top] = \bar{\Omega}_{22}^{-1}$. The random variable Y is independent of X_2 by definition, thus we get,

$$\begin{aligned}
&\mathbb{E}[(\boldsymbol{x}_1 - \mathbb{E}[\boldsymbol{x}_1])(\boldsymbol{x}_1 - \mathbb{E}[\boldsymbol{x}_1])^\top] \\
&= \Omega_{11}^{-1}\Omega_{12}\mathbb{E}[(\boldsymbol{x}_2 - \bar{\mu}_2)(\boldsymbol{x}_2 - \bar{\mu}_2)^\top]\Omega_{12}^\top\Omega_{11}^{-1} + \mathbb{E}[\eta\eta^\top] \\
&= \Omega_{11}^{-1}\Omega_{12}\bar{\Omega}_{22}^{-1}\Omega_{12}^\top\Omega_{11}^{-1} + \Omega_{11}^{-1}, \\
&\mathbb{E}[(\boldsymbol{x}_1 - \mathbb{E}[\boldsymbol{x}_1])(\boldsymbol{x}_2 - \mathbb{E}[\boldsymbol{x}_2])^\top] \\
&= \mathbb{E}[(\boldsymbol{x}_1 - (\Omega_{11}^{-1}\Omega_{12}(\mu_2 - \boldsymbol{x}_2) + \mu_1))(\boldsymbol{x}_2 - \bar{\mu}_2)^\top] \\
&= \mathbb{E}[-(\boldsymbol{x}_1 - \mu_1)(\boldsymbol{x}_2 - \bar{\mu}_2)^\top - \Omega_{11}^{-1}\Omega_{12}(\boldsymbol{x}_2 - \mu_2)(\boldsymbol{x}_2 - \bar{\mu}_2)^\top] \\
&= 0 - \Omega_{11}^{-1}\Omega_{12}\bar{\Omega}_{22}^{-1}, \\
&\mathbb{E}[(\boldsymbol{x}_2 - \mathbb{E}[\boldsymbol{x}_2])(\boldsymbol{x}_1 - \mathbb{E}[\boldsymbol{x}_1])^\top] = -\bar{\Omega}_{22}^{-1}\Omega_{12}^\top\Omega_{11}^{-1}, \\
&\mathbb{E}[(\boldsymbol{x}_2 - \mathbb{E}[\boldsymbol{x}_2])(\boldsymbol{x}_2 - \mathbb{E}[\boldsymbol{x}_2])^\top] = \bar{\Omega}_{22}^{-1}.
\end{aligned}$$

Using the following block matrix inversion formula on the covariance matrix blocks,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} \Omega_{11}^{-1}\Omega_{12}\bar{\Omega}_{22}^{-1}\Omega_{12}^\top\Omega_{11}^{-1} + \Omega_{11}^{-1} & -\Omega_{11}^{-1}\Omega_{12}\bar{\Omega}_{22}^{-1} \\ -\bar{\Omega}_{22}^{-1}\Omega_{12}^\top\Omega_{11}^{-1} & \bar{\Omega}_{22}^{-1} \end{bmatrix}^{-1}$$

we can obtain the updated normal distribution as

$$\mathcal{N}\left(\begin{bmatrix} \Omega_{11}^{-1}\Omega_{12}(\mu_2 - \bar{\mu}_2) + \mu_1 \\ \bar{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{12}^\top & \bar{\Omega}_{22} + \Omega_{12}^\top\Omega_{11}^{-1}\Omega_{12} \end{bmatrix}^{-1}\right)$$