Belief Consensus and Distributed Hypothesis Testing in Sensor Networks

Reza Olfati-Saber[1], Elisa Franco[2], Emilio Frazzoli[3], and Jeff S. Shamma[4]

[1] Dartmouth College, Thayer School of Engineeing, Hanover, NH 03755.
`olfati@dartmouth.edu` (the corresponding author)
[2] University of Trieste, Department of Electrical, Electronic, & Computer Engineering, Trieste, Italy. `efranco@univ.trieste.it`
[3] University of California, Los Angeles, Mechanical & Aerospace Engineering, Los Angeles, CA 90095. `frazzoli@ucla.edu`
[4] University of California, Los Angeles, Mechanical & Aerospace Engineering, Los Angeles, CA 90095. `shamma@seas.ucla.edu`

Summary. In this paper, we address distributed hypothesis testing (DHT) in sensor networks and Bayesian networks using the average-consensus algorithm of Olfati-Saber & Murray. As a byproduct, we obtain a novel belief propagation algorithm called Belief Consensus. This algorithm works for connected networks with loops and arbitrary degree sequence. Belief consensus allows distributed computation of products of n beliefs (or conditional probabilities) that belong to n different nodes of a network. This capability enables distributed hypothesis testing for a broad variety of applications. We show that this belief propagation admits a Lyapunov function that quantifies the collective disbelief in the network. Belief consensus benefits from scalability, robustness to link failures, convergence under variable topology, asynchronous features of average-consensus algorithm. Some connections between small-word networks and speed of convergence of belief consensus are discussed. A detailed example is provided for distributed detection of multi-target formations in a sensor network. The entire network is capable of reaching a common set of beliefs associated with correctness of different hypotheses. We demonstrate that our DHT algorithm successfully identifies a test formation in a network of sensors with self-constructed statistical models.

Key words: distributed hypothesis testing, multi-target tracking, Bayesian networks, average consensus, belief propagation, sensor networks, small-world networks

1 Introduction

Sensor networks are comprised of wireless networks of low-cost sensors that can be embedded in an environment for data gathering and monitoring purposes. Over the past decade, both sensor networks and ad-hoc networks of mobile agents have attracted many researchers from diverse backgrounds in computer science, robotics, signal processing, and control theory [20, 7, 10, 11, 5, 1, 6, 17, 14, 24, 28]. Information processing in sensor networks is closely related to traditional areas of research in systems and control such as *multi-target tracking* and *sensor fusion* back in 70's and 80's [3, 25, 4, 21, 30].

In a rather independent path, *belief networks* and graphical models were introduced in the pioneering work of Pearl [18] that turned into a powerful tool for statistical learning. The connections between the two areas of decentralized sensor fusion & target tracking and decentralized computation (not distributed) on Bayesian networks has been investigated by Rao & Durrant-Whyte [19] and more recently by Alanyali *et al.* [2] and Saligrama *et al.* [23]. An important issue that has been addressed is belief propagation in networks with loops (or cycles) [26, 29, 12] which was not known initially [18]. The objective stated in [12] to average numbers over a network using belief propagation is equivalent in nature to solving average-consensus problem [16] in an asynchronous mode [9].

Decentralized algorithms and distributed algorithms significantly differ in terms of their communication costs. In the decentralized algorithm proposed by Rao & Durrant-Whyte [19] , every node is allowed to talk to all other nodes and therefore the network has $O(n^2)$ number of links. As a result, a decentralized algorithm is *not* scalable. In contrast, in our distributed algorithm for hypothesis testing, each node only talks to a few of its neighbors. The total number of links is either $O(n)$, or $O(n\log(n))$ and in either case the algorithm in scalable in n. Hence, the fundamental between a decentralized system and a distributed system is that an all-to-all communication (or fully connected) topology is admissible in a decentralized system whereas it is unacceptable for a distributed system.

In this paper, our main objective is to solve the distributed hypothesis testing problem for sensors networks with applications to multi-target tracking and detection. The main features of interest are scalability and robustness of the algorithm to communication failures in the network. We demonstrate that *distributed hypothesis testing* can be posed as an *average-consensus problem* [16, 22] in terms of likelihood values of the nodes of the sensor network. As a byproduct of this algorithm, we obtain a novel belief propagation algorithm called *belief consensus algorithm*. Belief consensus works for loopy networks with arbitrary degree sequence. The convergence properties of this distributed algorithm is analyzed and a Lyapunov function is introduced that measures the *disbelief* in the network. The belief consensus algorithm allows distributed calculation of products of n probabilities that live in n different nodes. This enables distributed hypothesis testing that is scalable in comparison with a decentralized algorithm presented in [19].

To demonstrate the effectiveness of belief consensus, we present a detailed example of distributed detection of multi-target formations of $m \geq 2$ objects. Each sensor locally constructs statistical models of its observations that are later used for formation identification/detection. The sensors execute the belief consensus algorithm to reach a *common set of beliefs* representing the probability of correctness of multiple hypotheses. The consensus vector of beliefs is used to collectively find the most likely hypothesis (or formation) by the entire network of sensors.

An outline of the paper is as follows: Section 2 discusses the connections between consensus problems and belief propagation in Bayesian networks and contains our main results. A Lyapunov function for convergence of belief consensus is given in Section 3. Distributed detection of multi-target formations is discussed in Section 3. The simulation results are presented in Section 5. Finally, concluding remarks are stated in Section 6.

2 Belief Consensus on Bayesian Networks

Consider a sensor network with n sensors each making an observation $z(i)$. Let $Z = \{z(1), \dots, z(n)\}$ denote the set of observations/measurements of all sensors in the network. For example, each $z(i)$ can be the position of m moving targets in a *multi-target tracking* framework. In this paper, we are interested in solving a multi-hypothesis testing problem for sensor networks using a distributed algorithm that primarily relies on the average-consensus algorithm [16].

Similar to the framework used by Rao & Durrant-Whyte [19], calculation of *posterior* probabilities in a Bayesian network can be performed based on the following equation:

$$p(h_a|Z) = \alpha p(h_a) \prod_{i=1}^{n} p(z(i)|h_a) \tag{1}$$

under the assumption of independence of conditional probabilities (α is a normalization factor). The set of all hypotheses is $H = \{h_a\}_{a=1}^{n_h}$. We are interested in distributed computation of the following quantity that depends on the measurements of n sensors with a network topology $G = (V, E)$ that is an undirected graph:

$$Q = \prod_{i=1}^{n} p(z(i)|h_a) \implies Q' = \frac{1}{n}\log(Q) = \frac{1}{n}\sum_{i=1}^{n}\log(p(z(i)|h_a)) \tag{2}$$

Following the terminology of Pearl [18], let us refer to the conditional probability $\pi_i = p(z(i)|h_a)$ as the *belief* of agent i in the network. Defining the *likelihood of the belief* of agent i as $l_i := \log(\pi_i)$. Following the terminology of Pearl [18], let us call π_i the *belief of agent i* in the network. We have $Q' = \text{Ave}\{\log(\pi_i)\}_{i\in V} = \text{Ave}(l)$ and $Q = \exp(nQ') = \prod_i \pi_i$ where $\text{Ave}(\cdot)$ denotes the average $l = (l_1, \dots, l_n)$ is the vector of likelihoods. In other words, our objective is to calculate the product of beliefs using a distributed algorithm.

Let $E \subset V \times V$ be the set of edges of G. The set of *neighbors* of node i is defined as $N_i = \{j \in V : (i,j) \in E\}$. The *degree* of node i is denoted by $d_i = |N_i|$ (i.e. the number of neighbors of node i). The following average-consensus algorithm calculates Q' (and thus Q) in a distributed way on a network G:

$$\dot{x}_i = \sum_{j\in N_i}(x_j - x_i), \quad x_i(0) = \log(\pi_i) \tag{3}$$

The discrete-time *average-consensus algorithm* can be expressed as

$$x_i(t+1) = x_i(t) + \epsilon \sum_{j \in N_i} (x_j - x_i), \quad x_i(0) = \log(\pi_i) \tag{4}$$

for $t = 0, 1, 2, \ldots$ with $0 < \epsilon < 1/\Delta(G)$ that depends on the maximum node degree of the network $\Delta = \max_i |N_i|$. Based on the last iterative algorithm, the collective dynamics of consensus can be rewritten as $x(t+1) = (I - \epsilon L)x(t)$, or

$$x(t+1) = Px(t) \tag{5}$$

with $P = I - \epsilon L$ where L is the *graph Laplacian*. By definition, P is a *nonnegative matrix*. Moreover, due to the fact that $\mathbf{1} = (1, \ldots, 1)^T$ is the left and right eigenvector of L (i. e. $L\mathbf{1} = 0$ and $\mathbf{1}^T L = 0$), P becomes a *doubly-stochastic nonnegative matrix*. The inequality $0 < \epsilon < 1/\Delta(G)$ guarantees that P is a stable (or primitive) matrix based on Gershgorin's disk theorem.

In the following lemma, we show that all nodes asymptotically reach an agreement regarding Q' (and therefore Q) if the network is connected. The solution exponentially converges to a consensus state with a speed that is greater or equal to $\mu_2 = 1 - \epsilon\lambda_2$ with $\epsilon = 1/(1 + \Delta)$.

Lemma 1. *Consider a connected network of agents with topology G. Suppose every agent applies the consensus algorithm in* (4). *Then, the state of all agents exponentially converges to $\bar{x}(0) = (\sum_i x_i(0))/n$ with the least rate of $\mu_2 = 1 - \epsilon\lambda_2$ and average-consensus is asymptotically achieved.*

Proof. Let $\bar{x} = \text{Ave}(x)$ and note that $\bar{x}$ is an invariant quantity. Set $\tilde{x}_i = x_i - \bar{x}$ and note that $\tilde{x}(t+1) = P\tilde{x}(t)$. By definition, $\sum_i \tilde{x}_i = 0$ and thus $\tilde{x}$ is orthogonal to $\mathbf{1} = (1, \ldots, 1)^T$. This implies $\|\tilde{x}(t+1)\| = \|P\tilde{x}(t)\| \leq \mu_2 \|\tilde{x}(t)\|$ and as a result $\tilde{x}(t)$ exponentially converges to zero with a speed faster or equal to μ_2 and $\lim_{t\to\infty} x_i(t) = \bar{x}(0)$ for all i. See [16] for more details. □

Here is our main result on distributed computation of products of n conditional probabilities and derivation of a novel belief propagation algorithm:

Theorem 1. *Consider a connected (undirected) network of agents that exchange the likelihood of their beliefs $l_i = \log(\pi_i)$ according to the average-consensus algorithm in* (4). *Then, the following statements hold:*

i) The agents are capable of asymptotically reaching a product-consensus regarding the value of $Q = (\prod_{i=1}^{n} \pi_i)$ with $\pi_i > 0, \forall i$.

ii) The belief consensus algorithm for agreement in π_i's takes the form

$$\pi_i(t+1) = \pi_i(t)^{\beta_i} \prod_{j \in N_i} \pi_j(t)^{\gamma} \tag{6}$$

where $\beta_i = 1 - \gamma d_i > 0$ for all i and γ satisfies $0 < \gamma < 1/\Delta$ (d_i is the degree of node i and $\Delta = \max_i d_i$).

iii) $Q = \prod_{i=1}^{n} \pi_i$ is a invariant quantity along the solutions of (6) *and the group consensus value is $Q^{1/n}$ (i.e. $\pi_i \to Q^{1/n}$ as $t \to \infty$).*

Proof. Part i) follows directly from Lemma 1 and the identity $Q = \exp(nQ')$ where $Q' = \text{Ave}(l)$ is the average of node likelihoods. To prove part ii), let us define $\pi_i = \exp(x_i)$. Taking the exponential of both sides of the average-consensus algorithm in (4), we obtain a belief consensus algorithm in the form

$$\pi_i(t+1) = \pi_i(t) \left(\prod_{j \in N_i} \pi_j(t)/\pi_i(t) \right)^{\gamma} \tag{7}$$

by setting $\gamma = \epsilon$. Noting that $d_i = |N_i|$ and $\prod_{j \in N_i} \pi_i^{\gamma} = \pi_i^{\gamma d_i}$, equation (6) follows.

Part iii) follows from the product-consensus algorithm in the form (7) and the fact that in an undirected network, the ratio π_j/π_i corresponding to edge (i,j) cancels out with the ratio π_j/π_i due to edge (j,i). Hence, one obtains

$$\prod_{i=1}^{n} \pi_i(t+1) = \prod_{i=1}^{n} \pi_i(t) \left(\prod_{i<j} \frac{\pi_j(t)}{\pi_i(t)} \frac{\pi_i(t)}{\pi_j(t)} \right)^{\gamma} = \prod_{i=1}^{n} \pi_i(t),$$

or $Q(t+1) = Q(t)$. Given that the product-consensus algorithm in (6) converges, in the limit $\pi_i = \pi_j$ for all $i, j, j \neq i$ and $Q = \pi_i^n$ which means asymptotically the belief of all agents converges to $\pi_i = Q^{1/n}$. □

According to Theorem 1, we have arrived at a novel *belief propagation algorithm* given in (6) for distributed computation of products of beliefs of all agents in a network. We call the following algorithm belief consensus[5]:

Algorithm 1: *Belief Consensus* is an iterative belief propagation algorithm in the following form

$$\pi_i \leftarrow \pi_i \prod_{j \in N_i} (\pi_j/\pi_i)^{\gamma}. \tag{8}$$

Theorem 1 provides the convergece analysis of this algorithm. The belief consensus algorithm works for connected *networks with (or without) loops and arbitrary degree sequences*. A degree sequence for a graph is a vector $(d_1, d_2, \ldots, d_n)$ and an ”arbitrary degree sequence” means there are no restrictions on the degrees of all nodes to be equal or follow a particular pattern, or distribution. Traditionally, Pearl's original belief propagation algorithm is applicable to trees (or graphs without loops). The work in [2, 23] does not suffer from this restriction. However, for networks of finite size, the algorithm produces meaningful results for regular networks in which the degree of all nodes are equal (i.e. $d_1 = \ldots = d_n$). The belief consensus algorithm in (8) does not suffer from either of the aforementioned restrictions.

An important feature of Pearl's belief propagation algorithm is that it is *asynchronous*. Based on recent results by Hatano & Mesbahi [9] and Moreau [13], asynchronous implementations of consensus algorithms converge. Therefore, the belief consensus algorithm in *asynchronous mode* converges as well.

---

[5] This is due the fact that all agents asymptotically reach the same belief.

An advantage of equivalence between the average-consensus and belief consensus algorithms is that recently it was demonstrated by Olfati-Saber [15] that quasi-random *small-worlds* [27] are ”ultrafast” in dynamic diffusion of information. In the sense that consensus algorithms converge multiple orders of magnitude faster than they do on lattice-type regular networks. Apparently, this provides a means of designing the information flow of sensor networks in such a way that distributed inference and information processing can be carried out tremendously faster.

Note that the belief propagation algorithm of Pearl is fairly general and applicable to broader class of learning problems as long as the graphical models are trees. In comparison, belief consensus applies specifically to hypothesis testing with a wide range of applications and arbitrary graph topology. The main contribution is not belief propagation in loopy graphs, rather introducing a novel belief propagation algorithm that is robust to a changing topology and has a guaranteed speed of convergence characterized by algebraic connectivity (or λ_2) of the network.

Remark 1. (effects of uncertain network size) Suppose each node has an estimate $\hat{n}_i$ of the size of the network (n) with an error δ_i, i.e. $\hat{n}_i = n + \delta_i$. Assume $\delta_i/n \ll 1$. Then an estimate of the ith node of Q is

$$\hat{Q}_i = \exp(\hat{n}_i Q') = \exp(nQ')^{\hat{n}_i/n} = Q^{\nu_i}$$

with $\lim_{n\to\infty} \nu_i = \lim_{n\to\infty}(1 + \frac{\delta_i}{n}) = 1$. Therefore, for complex sensor networks $\hat{Q}_i \approx Q$ for all i. In case, $\delta_i = \delta$ is equal for all nodes, then $\hat{Q}_i = Q \exp(\frac{\delta}{n} Q') = Qc$ where c is a positive constant. This gives a new normalization constant $\beta = c\alpha$ in equation (1) and has no effect on hypothesis testing.

3 Lyapunov Function for Belief Consensus

A unique advantage of the equivalence between average-consensus in likelihoods in (4) and belief consensus in (8) is that a *Lyapunov function* is available for convergence of the average-consensus algorithm in discrete-time that is called the *disagreement function* [16] and is defined by

$$W(x) = (x - \bar{x}\mathbf{1})^T (x - \bar{x}\mathbf{1}). \tag{9}$$

Given the change of coordinates $x_i = \log(\pi_i)$, one obtains the following *collective disbelief function*

$$\varphi(\pi) = \exp(W(x)) \geq 1 \tag{10}$$

where $\pi = (\pi_1, \dots, \pi_n)^T$. The following theorem explicitly gives this function.

Theorem 2. *The disbelief function is the product of powers of beliefs with exponents that add up to zero, i.e. in the form* $\varphi(\pi) = \prod_i \pi_i^{\nu_i}$ *where* $\nu_i = \log(\pi_i) - (1/n)\log(Q)$ *is likelihood deviation of agent* i*. Furthermore,* $\varphi(\pi)$ *takes its minimum value of* 1 *when all agents have the same belief.*

Proof. First, note that by definition of Q, $\sum_i \nu_i = 0$. The function W can be written as

$$W(x) = \sum_i (x_i - \bar{x})^2 = \sum_i (\log(\pi_i) - 1/n \log(Q))^2$$

Thus

$$\begin{aligned}
\varphi(\pi) &= \prod_i \exp\{(\log(\pi_i) - 1/n \log(Q))^2\} && (11)\\
&= \prod_i \exp(\nu_i \log(\pi_i / Q^{1/n})) && (12)\\
&= \prod_i (\pi_i / Q^{1/n})^{\nu_i} = \left(\prod_i \pi_i^{\nu_i}\right) / Q^{(\sum_i \nu_i)/n} && (13)\\
&= \prod_i \pi_i^{\nu_i} && (14)
\end{aligned}$$

The last equality is due to $\sum_i \nu_i = 0$. Since $\varphi = \exp(W)$ and $W \geq 0$, φ takes its minimum at 1 when all agents have the same belief, i.e. $\log(\pi) = 1/n \log(Q)$ for all i. □

Remark 2. For two agents 1 and 2, the total disbelief can be stated as

$$(\pi_1/\pi_2)^{(\log(\pi_2) - \log(\pi_2))/2}.$$

In *Bayesian networks with variable topology*[6], the disbelief function can be effectively used as a common Lyapunov function in a similar way to the role of disagreement function in consensus on switching networks [16]. The feasibility of convergence analysis for belief consensus on switching Bayesian networks increases the hope for development of more general belief propagation algorithms for networks with variable interconnections. This framework could possibly have an instrumental role in understanding and modeling real-life biological and social networks with changing structural interactions.

4 Distributed Detection of Multi-Target Formations

We consider the problem of distributed detection of multi-target formations using a sensor network within a multi-target tracking framework. Each formation of the targets corresponds to a hypothesis h_a. We consider $n_h = 4$ types of (unperturbed) formations of targets and accordingly four hypotheses $a = 1, \ldots, n_h$. An example of the network of $n = 20$ sensors performing this distributed detection task is shown in Fig. 1.

---

[6] To the best of our knowledge, there exist no results on belief propagation on networks with changing topologies.

Remark 3. The term ”formation” here applies to an arrangement of physical objects. In complex social networks, *formations* represent *intent* or *behavior* of a group of agents. The framework discussed here is equally applicable to detection of intent of a group of *adversarial agents* in games or social systems. The application of the theoretical framework presented in this paper in social networks is the subject of ongoing research.

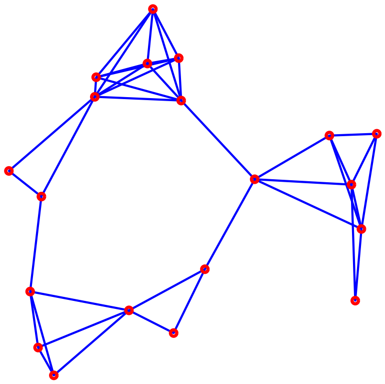


Fig. 1. A sensor network with $n = 20$ nodes.

4.1 Sensing Model of Sensors

Each sensor in a network with communication topology G observes a noisy set of measurements of the positions of m targets. The formations used in this paper all consists of $m = 12$ targets and are shown in Fig. 2. Let $q_j = (q_{j1}, q_{j2})^T \in \mathbb{R}^2$ be the position of the jth target, then the ith sensor observes a noisy version of the position of all targets that is an m-vector $z(i) = (y_1(i), \dots, y_m(i))$ in the form:

$$y_j(i) = q_j + w_j(i); \tag{15}$$
$$w_{j1}(i) = r_i \cos(\theta_i)); \tag{16}$$
$$w_{j2}(i) = r_i \sin(\theta_i). \tag{17}$$

where $w_j(i)$ is the measurement noise of the ith sensor for observation of the jth target. The quantities r_i and θ_i are random variables with uniform distribution over the intervals $[0, \bar{r}_i]$ and $[-\pi, \pi]$.

4.2 Feature Vectors for Multi-Target Formation Detection

We define a set of *features* for multi-target formations that are functions of the position of all targets in a formation but do not depend on ordering of the targets (i.e.

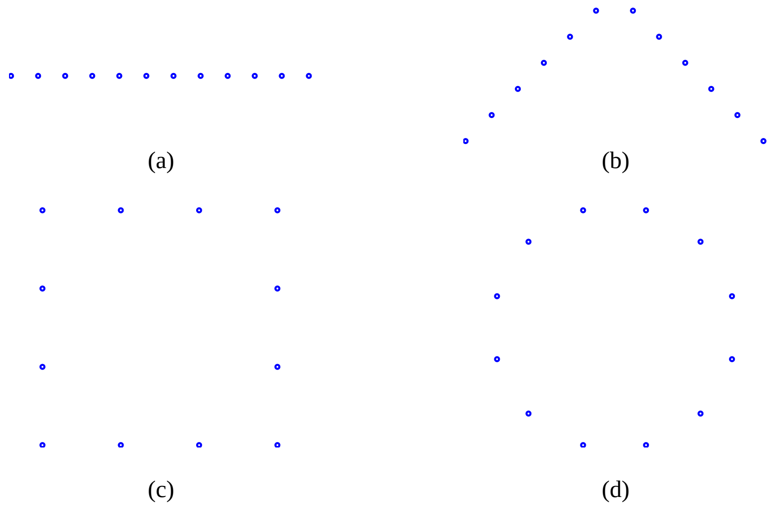


Fig. 2. Multi-target formations: (a) line formation, (b) V-formation, (c) square formation, and (d) circle formation.

the features are invariant under permutation of indices of the target). In addition. all the features are invariant with respect to rotation and translation of all targets. Let z be the observed position of m targets by some sensor (the index of the sensor is unimportant in this discussion). Define the *radius* of formation z as

$$\rho(z) = \max_{j \in 1,\dots,m} \|z(j) - \bar{z}\|; \ \bar{z} = \frac{1}{m}\sum_{j=1}^{m} z(j).$$

Given the radius $\rho(z)$, one can define a set of targets that are *neighbors* of each other using

$$N(\rho) = \{i, j : \|z(j) - z(i)\| \leq \rho\}.$$

Moreover, let $J = [z(1)|z(2)|\cdots|z(m)]$ denote the $2 \times m$ matrix of coordinates of points in the formation. Here are the five chosen features in the form $x = f(z)$:

$$x_1 = (\sum_{j=1}^{m} \|z(i)\|^2)^{1/2} \tag{18}$$

$$x_2 = \lambda_{max}(JJ^T)^{1/2} \tag{19}$$

$$x_3 = |N(\rho(z))| \tag{20}$$

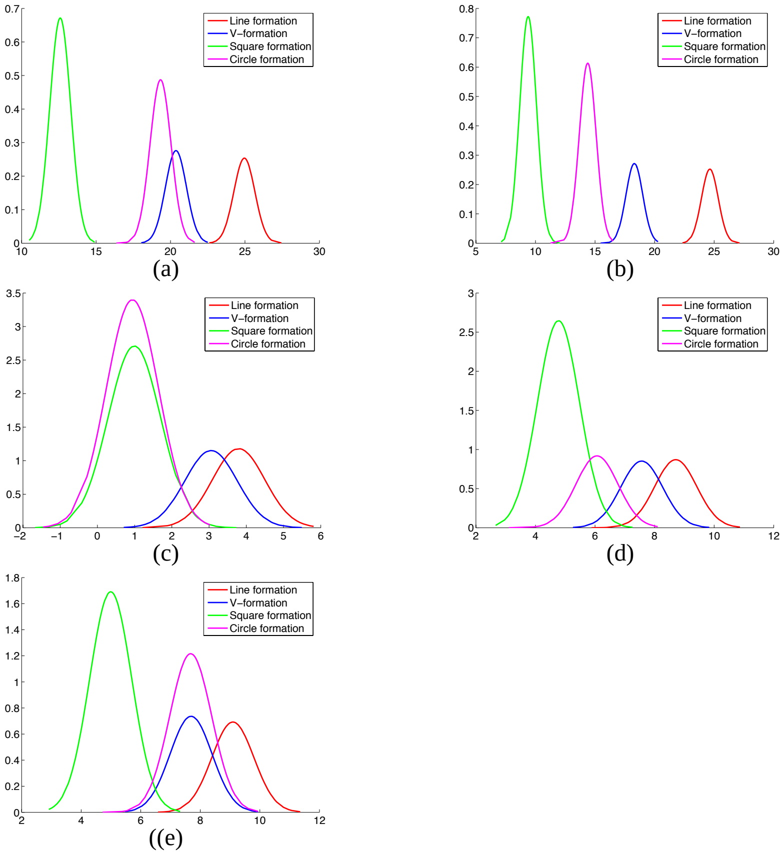
$$x_4 = |N(2\rho(z))| \tag{21}$$

$$x_4 = |N(4\rho(z))| \tag{22}$$

where $|\cdot|$ denotes the number of elements of a set.

4.3 Learning Statistical Models of Formations

Every sensor represents a formation using its feature vector $x = f(z)$ and based on $n_t = 100$ observations (training data) creates Gaussian distribution of the features x_k, $k = 1, \ldots, n_f = 5$. This can be done by measures mean and variance of the observed data as $\mu_k(i), \sigma_k^2(i)$ for sensors $i = 1, \ldots, n$. An example of these distributions is shown in Fig. 3.



(a) (b)

(c) (d)

((e)

Fig. 3. Gaussian distribution of different features: (a) x_1, (b) x_2, (c) x_3, (d) x_4, and (e) x_5.

4.4 Multi-Target Formation Detection

In the detection stage, we would like to use a maximum a-posterior (MAP) estimator that first calculates $p(h_a|Z)$ for $a = 1, \ldots, 4$ using belief consensus algorithm and then uses the following recognition step:

$$a^* = \arg\max_a p(h_a|Z).$$

For doing so, first we need to relate the statistics of the features of a formation to its prior probability $p(z_i|h_a)$. This can be greatly simplified by assuming that all features are statistically independent, i.e.

$$p(z(i)|h_a) := \prod_{k=1}^{n_f} p(x_k(i)|h_a) \tag{23}$$

where $x(i) = f(z(i))$ is the feature vector of the observation of the ith node. Regardless of the node index, we assume that each feature has a Gaussian distribution. As a result, the probability that a feature x_k is in an interval of length $\delta \ll 1$ can be approximated as

$$p(x_k|h_a) \approx \delta \times \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_k-\mu_k)^2}{\sigma_k^2}\right).$$

Since δ is a positive constant, it will be eventually absorbed in the normalization constant of posterior probability $p(h_a|Z)$ and therefore it can be dropped. Hence, we apply the following approximation for the kth feature calculated locally by every sensor according to sensor's statistical model of mean and variance of n_t previously observed data:

$$p(x_k|h_a) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_k-\mu_k)^2}{\sigma_k^2}\right). \tag{24}$$

for the kth feature calculated locally by every sensor according to sensor's statistical model of mean and variance of n_t previously observed multi-target formations.

4.5 Belief Consensus

Assuming that all n_h formations are equally likely to occur, we obtain $p(h_a) = 1/n_h$. Based on the procedure given in the preceding section, each sensor can calculate its n_h-dimensional vector of *beliefs* $\pi_i(a) = p(z_i|h_a)$. A MAP estimator only requires this knowledge that can be calculated either directly using belief consensus in (8), or indirectly using average-consensus algorithm in (4) (in terms of likelihood of beliefs).

5 Simulation Results

Consider a network of $n = 20$ sensor that are distributed randomly in a square region as shown in Fig. 1. The objective is to detect a test formation after learning the statistical features of training data of $n_h = 4$ formations shown in Fig. 2.

Each sensor observes a randomly perturbed version of the formation of $m = 12$ targets that are spaced approximately at a distance of $d = 2$.The perturbation radius of sensors monotonically increases by their index as $\bar{r}_i = 0.1 + .02i$ to create varying degrees of uncertainty. All sensors as well as the sensor network as a whole manage to correctly identify the test formation. The sensors asymptotically reach a *common set of beliefs* $\{\pi_i(a)\}_{a=1}^{n_h}$ that allow them to choose the most likely hypothesis with maximum likelihood.

The result of distributed detection of multi-target formations for a square formation ($a = 3$) are presented in Fig. 4. Apparently, the winning hypothesis is the one that its likelihood is negative but near zero (giving a relatively high common belief by all agents.

We also tested whether a network of sensors that is trained with formations taht have $m = 12$ targets is capable of identifying formations with less or more targets. It turns out that the answer is affirmitive for small changes in the number of targets.

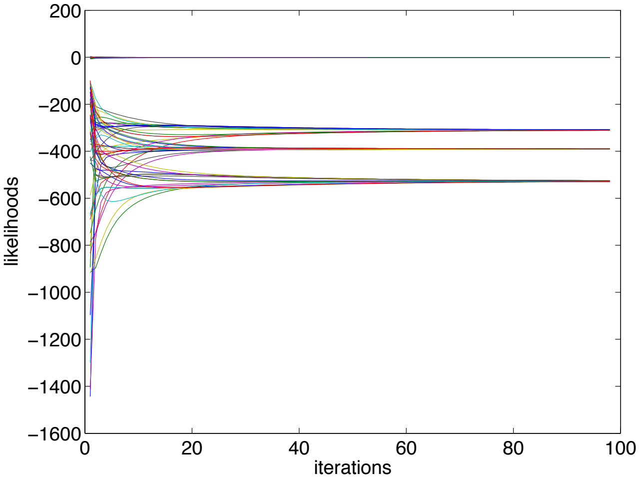


Fig. 4. Evolution of likelihood of $n = 20$ nodes of a sensor network for four hypotheses based on belief consensus.

6 Conclusions

A novel belief propagation algorithm called belief consensus was introduced for distributed computation of products of beliefs of n nodes of a sensor network. This computation is instrumental in distributed hypothesis testing in Bayesian networks. Belief consensus is a byproduct of the average-consensus algorithm in [16]. This belief propagation algorithm works for connected networks with loops/cycles and arbitrary degree sequence. A Lyapunov function was given for belief consensus in the form of collective disbelief of all agents. The disbelief function has a potentially important role in convergence analysis of belief consensus in Bayesian networks with variable topology. It was discussed that belief consensus (due to its connection to average-consensus) converges multiple orders of magnitude faster on quasi-random small-world networks compared to regular lattices or highly clustered small-worlds [15].

A detailed example was provided for distributed detection of multi-target formations using a network of sensors. Belief consensus was used to reach a common set of beliefs associated with correctness of different hypotheses. Then a MAP estimator was used to identify the formation/hypothesis with maximum likelihood. All agents could use a local MAP estimator after they reach a consensus with other agents regarding their set of beliefs. It was demonstrated that belief consensus successfully identifies a test formation in a network of sensors with individually built statistical models of previously observed multi-target formations. Recently, belief consensus has been applied to *distributed fault diagnosis* in network embedded systems [8].

Acknowledgments

This research was supported in part by ARO under the grant #W911NF–04–1–0316, AFOSR MURI grant #F49620-02-1-0325, and NSF ITR grant #0325716. This work was performed while the first author was a visiting scientist at the University of California, Los Angeles.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pages 102–114, August 2002.
2. M. Alanyali, V. Saligrama, O. Savas, and S. Aeron. Distributed Bayesian hypothesis testing in sensor networks. *Proceedings of the 2004 American Control Conference*, 6:5369–5374, 2004.
3. Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Trans. on Automatic Control*, 23(4):618–626, 1978.
4. Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*. Academic Press, Boston, MA, 1988.
5. M. Chu, H. Haussecker, and F. Zhao. Scalable information-deriven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High-Performance Computing Applications*, 16(3):293–313, 2002.
6. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. on Robotics and Automation*, 20(2):243–255, April 2004.
7. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. *Proc. of Mobile Computing and Networking*, pages 263–270, 1999.
8. E. Franco, R. Olfati-Saber, T. Parisini, and M. M. Polycarpou. Distributed fault diagnosis using sensor networks and consensus-based filters. *American Control Conference (submitted)*, June 2006.
9. Y. Hatano and M. Mesbahi. Agreement over random networks. *Proceedings of the 43rd Conference on Decision and Control*, 2:2010–2015, Dec. 2004.

10. W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *Proc. of Mobile Computing and Networking*, pages 174–185, 1999.
11. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *Proceedings of IEEE Infocom*, 3:1380–1387, April 2001.
12. C. C. Moallemi and B. Van Roy. Consensus propagation. *Preprint (submitted)*, June 2005.
13. L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. on Automatic Control*, 50(2):169–182, Feb 2005.
14. R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. *Proceedings of the 44th Conference on Decision and Control*, Dec 2005.
15. R. Olfati-Saber. Ultrafast Consensus in Small-World Networks. *Proceedings of the 2005 American Control Conference*, pages 2371–2378, June 2005.
16. R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *The IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sep. 2004.
17. R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fudion. *Proceedings of the 44th Conference on Decision and Control*, Dec 2005.
18. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
19. B. S. Y. Rao and Durrant-Whyte. A decentralized Bayesian algorithm for identification of tracked targets. *IEEE Trans. on Systems, Man, and Cybernetics*, 23(6):1683–1698, 1993.
20. B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *Int. Journal of Robotics Research*, 12(1):20–44, Feb 1993.
21. D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, 1979.
22. R. O. Saber and R. M. Murray. Consensus protocols for networks of dynamic agents. *Proceeding of the 2003 American Control Conference*, 2:951–956, June 2003.
23. V. Saligrama, M. Alanyali, and O. Savas. Asynchronous distributed detection in sensor networks. *preprint*, 2005.
24. D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. *Fourth International Symposium on Information Processing in Sensor Networks*, pages 133–139, April 2005.
25. J. L. Speyer. Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem. *IEEE Trans. on Automatic Control*, 24(2):266–269, Feb 1979.
26. S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. *Proceedings of the 40th Allerton Conference on Communication, Control, and Computing*, 2002.
27. D. J. Watts and S. H. Strogatz. Collective dynamics of ’small-world’ networks. *Nature*, 393:440–442, June 1998.
28. L. Xiao, S. Boyd, and S. Lall. A scheme for asynchronuous distributed sensor fusion based on average consensus. *Fourth International Symposium on Information Processing in Sensor Networks*, April 2005.
29. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief ropagation algorithm. Technical Report 2004–040, MERL, 2004.
30. B. Zhou and N. K. Bose. Multi-target tracking in clutter: fast algorithms for data associations. *IEEE Trans. on Aerospace and Electronic Systems*, 29(2):352–363, 1993.