Distributed combination of belief functions

Thierry Denœux[a,b,c]

[a] *Université de technologie de Compiègne, CNRS, Heudiasyc, Compiègne, France*
[b] *Shanghai University, UTSEUS, Shanghai, China*
[c] *Institut universitaire de France, Paris, France*

Abstract

We consider the problem of combining belief functions in a situation where pieces of evidence are held by agents at the node of a communication network, and each agent can only exchange information with its neighbors. Using the concept of weight of evidence, we propose distributed implementations of Dempster's rule and the cautious rule based, respectively, on average and maximum consensus algorithms. We also describe distributed procedures whereby the agents can agree on a frame of discernment and a list of supported hypotheses, thus reducing the amount of data to be exchanged in the network. Finally, we show the feasibility of a robust combination procedure based on a distributed implementation of the random sample consensus (RANSAC) algorithm.

Keywords: Dempster-Shafer theory, evidence theory, consensus, information fusion, uncertain reasoning.

1. Introduction

Since its development by Shafer [35] following Dempster's seminal work on statistical inference [8], Dempster-Shafer (DS) theory has been widely used as a formal framework for uncertain reasoning [10, 11]. In the past thirty years, it has been used extensively in a large number of applications including information fusion [7, 33, 44], classification [14, 13], clustering [15], scene perception [47], etc.

DS theory is essentially a *theory of evidence*: it consists in representing elementary pieces of evidence pertaining to a question of interest using belief functions, i.e., completely monotone set functions [35, 11], and pooling them using some appropriate combination rule. Dempster's rule, also referred to as the product-intersection rule, occupies a central position in DS theory; it allows us to combine pieces of evidence that are both reliable and independent [35]. However, in real-world applications, it is not always possible to break down the available evidence into independent pieces, and the complete reliability that is called for in the basic theory is often an idealization. For that reason, alternative combination rules

Email address: `Thierry.Denoeux@utc.fr` (Thierry Denœux)

have been developed over the years to handle highly conflicting [19, 31, 41, 4] or dependent pieces of evidence [9, 16, 5, 3].

Most implementations of Dempster's rule assume a centralized fusion scheme in which a single agent receives pieces of evidence from several sources and combines them. There are, however, a growing number of applications in which several agents independently collect evidence and exchange information via a static or dynamic communication network. In this case, no single agent holds the totality of the evidence, and agents typically can only access the information held by their neighbors in the communication graph. As typical applications, we can mention target classification in sensor networks [25] and information fusion in multi-robot systems [32]. For instance, El Zoghby et al. [20] describe a *collaborative perception* application in which a fleet of intelligent vehicles is equipped with sensors and communicates through an ad hoc network (see also, e.g. [28]). The overall objective of this application is to enhance each vehicle's perception and situation awareness of a complex dynamic traffic scene through the multiplicity of sensors and the communication capabilities of the agents. Typically, each agent perceives a number of objects and collects sensor information about them. Object association and classification then have to be performed using *distributed algorithms*, which allow the agents to exchange information locally with their neighbors in the network and to construct, collectively, a shared representation of the environment. Distributed algorithms for object association are described in [32]. Here, we assume that the agents have already agreed on some questions of interest (such as the class of matched objects) and we focus on the distributed combination of evidence assumed to be represented by belief functions.

A first approach to this problem has been proposed in [26], in which the authors proposed a distributed implementation of Dempster's rule based on the combination of commonality functions, an alternative representation of belief functions. However, a problem with this approach is that the quantity of information to be exchanged in the network grows exponentially with the number of hypotheses. Furthermore, this approach does not easily extend to other combination rules. In this paper, we propose an alternative approach based on the combination of *weights of evidence*. As opposed to commonalities, weights of evidence are usually specified for a small number of hypotheses, which considerably reduces the amount of communications as well as the complexity of computations at each node. Furthermore, the same approach can be used to develop a distributed implementation of the cautious rule, an alternative to Dempster's rule making it possible to combine dependent items of evidence. Finally, we propose a distributed RANSAC algorithm allowing the robust combination of belief functions to account, for instance, for wrong associations or faulty sensors.

The rest of this paper is organized as follows. The necessary background on DS theory is first recalled in Section 2. The distributed implementations of the product-intersection and cautious rules are then described in Section 3. Finally, the robust combination procedure is presented in Section 4, and Section 5 concludes the paper.

2. Background on belief functions

In this section, background knowledge on DS theory is summarized to make the paper self-contained. Basic notions are first recalled in Section 2.1. Weights of evidence and the cautious rule of combination are then introduced, respectively, in Sections 2.2 and 2.3.

2.1. Basic notions

We consider a question of interest having one and only one answer in a finite set Ω called the *frame of discernment*. A *mass function* on Ω is a mapping $m : 2^\Omega \to [0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1.$$

If $m(\varnothing) > 0$, mass function m is said to be *subnormal*, otherwise it is said to be *normalized*. A subnormal mass function such that $m(\varnothing) < 1$ can be transformed to a normalized one m^* defined by $m^*(\varnothing) = 0$ and

$$m^*(A) = \frac{m(A)}{1 - m(\varnothing)} \tag{1}$$

for all nonempty subset $A \subseteq \Omega$. This operation is called *normalization*. Mass functions are usually assumed to be normalized, but unnormalized mass functions sometimes appear as intermediate results in calculations (see below).

A mass function m represents a *piece of evidence*, and $m(A)$ represents the probability that the evidence tells us that the truth lies in A, and nothing more specific. The subsets A such that $m(A) > 0$ are called the *focal sets* of m. A mass function is said to be *consonant* if, for any two focal sets A and B, we have either $A \subset B$ or $B \subset A$. It is said to be *Bayesian* if all its focal sets are singletons. Finally, the *vacuous* mass function $m_?$ verifies $m_?(\Omega) = 1$; it represents complete ignorance.

A *belief function* $Bel : 2^\Omega \to [0, 1]$ can be computed from a *normalized* mass function m^* as

$$Bel(A) = \sum_{B \subseteq A} m^*(B),$$

for all $A \subseteq \Omega$. The quantity $Bel(A)$ is interpreted as the total degree of support given to the proposition that the truth lies in A, taking into account the support given to A and to all of its subsets. Obviously, $Bel(\varnothing) = 0$ and $Bel(\Omega) = 1$. A related notion is that of *plausibility function*, defined as

$$Pl(A) = \sum_{B \cap A \neq \varnothing} m^*(B) = 1 - Bel(\overline{A}),$$

where $\overline{A}$ denotes the complement of A. The quantity $Pl(B)$ reflects the lack of support given to the proposition that the truth does *not* lie in A.

Two mass functions m_1 and m_2 on Ω representing *independent* items of evidence can be combined by the *conjunctive sum* operation $\bigcirc\hspace{-0.75em}\cap$ [38] defined as

$$(m_1 \bigcirc\hspace{-0.75em}\cap\, m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \tag{2}$$

for all subset A of Ω. Obviously, $m_1 \textcircled{\cap} m_2$ may be subnormal, even if m_1 and m_2 are normalized. The quantity

$$\kappa(m_1, m_2) = (m_1 \textcircled{\cap} m_2)(\varnothing)$$

is called the *degree of conflict* between m_1 and m_2. Normalizing $m_1 \textcircled{\cap} m_2$ yields a normalized mass function $m_1 \oplus m_2$, called the *orthogonal sum* of m_1 and m_2 and defined as

$$m_1 \oplus m_2 = (m_1 \textcircled{\cap} m_2)^* . \tag{3}$$

The combination rule $\oplus$ defined by (3) is called *Dempster's rule of combination*, or the *product-intersection* rule. Both the unnormalized conjunctive rule $\textcircled{\cap}$ and Dempster's rule $\oplus$ are commutative and associative, and the vacuous element $m_?$ is the neutral element of both operators. An important property is that, to combine several mass functions by Dempster's rule, we can use the unnormalized combination operator (2) and normalize only at the end, i.e., for n arbitrary mass functions $m_1, \ldots, m_n$,

$$m_1 \oplus \ldots \oplus m_n = (m_1 \textcircled{\cap} \ldots \textcircled{\cap} m_n)^* . \tag{4}$$

Dempster's rule can also be computed using the *commonality function.* The commonality function $Q : 2^\Omega \to [0, 1]$ associated to a mass function m is defined by

$$Q(A) = \sum_{B \supseteq A} m(B), \tag{5}$$

for all $A \subseteq \Omega$. Conversely, m can be recovered from Q using the following formula,

$$m(A) = \sum_{B \supseteq A} (-1)^{|B|-|A|} Q(B) \tag{6}$$

for all $A \subseteq \Omega$. If Q_1 and Q_2 are the commonality functions associated with two mass functions m_1 and m_2, then the commonality function $Q_1 \textcircled{\cap} Q_2$ associated with $m_1 \textcircled{\cap} m_2$ is the product of Q_1 and Q_2:

$$Q_1 \textcircled{\cap} Q_2 = Q_1 \cdot Q_2. \tag{7}$$

Dempster's rule can, thus, be implemented by computing the commonality functions using (5), multiplying them point-wise, converting the result back to a mass function using (6), and renormalizing using (1). However, the complexity of this procedure is $O(2^K)$, where K is the cardinality of the frame of discernment. In contrast, masses are often assigned to a small number of focal sets, which makes the computation of Dempster's rule using (2)-(3) much more efficient.

2.2. Weights of evidence and separability

In practical applications, an elementary piece of evidence about a question of interest often comes as a nonempty set $A \subseteq \Omega$ of possible answers, and a degree of support $s \in [0, 1)$ for that set. Mathematically, such a piece of information can be formalized as a *simple mass function* m of the form

$$m(A) = s, \quad m(\Omega) = 1 - s. \tag{8}$$

Given two simple mass functions m_1 and m_2 with the same focal set A and degrees of support s_1 and s_2, their orthogonal sum is

$$(m_1 \oplus m_2)(A) = 1 - (1 - s_1)(1 - s_2) \tag{9a}$$

$$(m_1 \oplus m_2)(\Omega) = (1 - s_1)(1 - s_2). \tag{9b}$$

Denoting the simple mass function (8) by A^w, where $w = -\ln(1-s)$ is the *weight of evidence*[1] for A as defined by Shafer [35, page 77], Eq. (9) becomes

$$A^{w_1} \oplus A^{w_2} = A^{w_1+w_2},$$

i.e., weights of evidence add up when combining simple mass functions with the same focus. The vacuous mass function can be written as A^0 for any $A \subseteq \Omega$.

A normal mass function m is said to be *separable* [35] if it is the orthogonal sum of simple mass functions; it can be written as

$$m = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w(A)}, \tag{10}$$

where $w(A)$ is the weight of evidence for subset A. As a consequence of Theorem 5.2 in [35], this decomposition is unique and the weights $w(A)$ are all finite as long as $m(\Omega) > 0$ (m is then said to be *nondogmatic*). Most mass functions used in practice are separable. This is the case, in particular, for consonant mass functions [9].

If m is nondogmatic, the weights in (10) can be computed from the commonality function by the following equation [40]:

$$w(A) = \sum_{B \supseteq A} (-1)^{|B|-|A|} \ln Q(B), \tag{11}$$

for all $A \in 2^\Omega \setminus \{\emptyset, \Omega\}$. This equation provides a test to determine if a normalized mass function m is separable: the property holds iff the weights $w(A)$ computed from (11) are all positive or equal to zero. For an arbitrary nondogmatic mass function, we can still define a corresponding *weight assignment* as the mapping from $2^\Omega \setminus \{\emptyset, \Omega\}$ to $\mathbb{R}$ expressed by (11).

Let w_1 and w_2 be the weight assignments associated to two mass functions m_1 and m_2, and let $w_{1\oplus 2}$ be the weight assignment corresponding to $m_1 \oplus m_2$. As a consequence of (7) and (11), we have

$$w_{1\oplus 2} = w_1 + w_2, \tag{12}$$

i.e., in the weight representation, Dempster's rule is just addition. Given the weight assignment w, the corresponding mass function can be recovered from (10), but $A^{w(A)}$ is only a valid simple mass function if $w(A) \geq 0$. If $w(A) < 0$, the notation $A^{w(A)}$ can still be used to designate a "generalized simple mass function" [40], defined as a mapping that assigns

[1] In [9], following [40], we used the term "weight" for $\exp(-w)$, a quantity called "diffidence" in [17]. As we will see, the additivity property is central in our analysis: we thus stick to Shafer's terminology and notation in this paper.

Table 1: Two mass functions with the corresponding commonality functions and weight assignments, and their orthogonal sum. The weight assignment of the orthogonal sum $m_{1\oplus 2} = m_1 \oplus m_2$ is the sum of the weight assignments of m_1 and m_2.

	$\varnothing$	$\{a\}$	$\{b\}$	$\{a,b\}$	$\{c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,b,c\}$
m_1	·	0.5	·	0.3	·	0.1	·	0.1
Q_1	1	1	0.4	0.4	0.2	0.2	0.1	0.1
w_1		0.22	·	1.39	·	0.69	·	
m_2	·	·	·	·	·	0.4	0.2	0.4
Q_2	1	0.8	0.6	0.4	1	0.8	0.6	0.4
w_2		·	·	·	−0.182	0.69	0.41	
$m_{1\oplus 2}$	·	0.58	0.067	0.13	0.022	0.13	0.022	0.044
$Q_{1\oplus 2}$	1	0.89	0.267	0.18	0.222	0.18	0.067	0.044
$w_{1\oplus 2}$		0.22	·	1.39	−0.182	1.39	0.41	

a negative mass $s = 1 - e^{-w(A)}$ to A, and a mass $e^{-w(A)} > 1$ to Ω. Such mappings are not proper mass functions and they do not have a simple interpretation, but they can still be formally combined by Dempster's rule using (2)-(3). This notational trick makes it possible to recover m from w using (10) for any arbitrary nondogmatic mass function m.

Example 1. *Let $\Omega = \{a, b, c\}$ be a frame of discernment with three elements. Table 1 shows two mass functions m_1 and m_2 with their corresponding commonality functions and weight assignments, as well as the result of the combination of m_1 and m_2 by Dempster's rule. We can see that m_1 is separable (the corresponding weights are all positive), whereas m_2 is not (as $w_2(\{c\}) < 0$). A negative weight of evidence can be interpreted as the "retraction" of some evidence [40, 17]. The weight assignment $w_{1\oplus 2}$ of $m_{1\oplus 2} = m_1 \oplus m_2$ is the sum of the weight assignments w_1 and w_2. We remark that $m_{1\oplus 2}$ is not separable.*

Even though the mass functions of Example 1 are defined over a very small frame, we can already observe that the Q-representation is less parsimonious than the w-representation. This is due to the fact that the Q-representation is *global*: if some positive mass $m(A) > 0$ is assigned to some focal set A, all subsets of A will have a strictly positive commonality. In contrast, the weight-of-evidence representation is *local*: the combination with a simple mass function focused on a subset A only changes the weight of A. As a consequence, weights of evidence are usually assigned to only a few subsets of the frame of discernment supported by the evidence (called *supported sets* in the rest of this paper), and the number of nonzero weights typically remains small when combining several mass functions by Dempster's rule.

2.3. Cautious rule of combination

In DS reasoning, it is often useful to compare the "information content" of two mass functions. This allows us, in particular, to find the *least-committed* mass function compatible with a set of constraints induced by the available evidence, a principle known as the *Least-Commitment Principle* (LCP) [39]. Several informational orderings generalizing set inclusion have been proposed. For instance, the *q-ordering* [18] is defined by

$$m_1 \sqsubseteq_q m_2 \ \text{ iff } \ \forall A \subseteq \Omega,\ Q_1(A) \leq Q_2(A).$$

In particular, if there exists a mass function m such that $m_1 = m_2 \oplus m$ and m is not conflicting with m_2, i.e., $\kappa(m_2, m) = 0$, then m_1 is q-more committed than m_2; in this case we have obviously $Q_1 = Q_2 \cdot Q \leq Q_2$. Being the orthogonal sum of m_2 and another mass function m, m_1 then clearly contains more information than m_2.

Another informational ordering, introduced in [9], is based on the comparison between weight assignments: for any two nondogmatic mass functions m_1 and m_2,

$$m_1 \sqsubseteq_w m_2 \ \text{ iff } \ \forall A \subseteq \Omega,\ w_1(A) \geq w_2(A),$$

i.e., m_1 is w-more committed than m_2 if and only if it assigns larger weights of evidence to all hypotheses. An alternative interpretation is as follows: m_1 is w-more committed than m_2 iff there exists a *separable* mass function m such that $m_1 = m_2 \oplus m$.

Let us now assume that we have two mass functions m_1 and m_2 from two sources that are *not independent*, and we wish to combine them conjunctively. The combined mass function m_{12} should be more committed than both m_1 and m_2. Let $\mathcal{S}(m_1)$ and $\mathcal{S}(m_2)$ denote the sets of mass functions that are more committed than, respectively, m_1 and m_2. We thus require that $m_{12} \in \mathcal{S}(m_1) \cap \mathcal{S}(m_2)$. According to the LCP, the least committed mass function in the set $\mathcal{S}(m_1) \cap \mathcal{S}(m_2)$, according to some ordering $\sqsubseteq_x$, should be selected, if it exists. The existence of a least-committed element is not always guaranteed. If m_1 and m_2 are consonant and $x = q$, then the solution is the consonant mass function m_{12} such that $Pl_{12}(\{\omega\}) = Pl_1(\{\omega\}) \wedge Pl_2(\{\omega\})$ for all $\omega \in \Omega$, where $\wedge$ denotes the minimum operator. However, the consonance condition is quite strict. For the w-ordering, the least-committed mass function in $\mathcal{S}(m_1) \cap \mathcal{S}(m_2)$ exists and is unique for any nondogmatic mass function [9]; it is given by

$$m_1 \owedge m_2 = \bigoplus_{\emptyset \neq A \subset \Omega} A^{w_1(A) \vee w_2(A)},$$

where $\vee$ denotes the maximum operator. The $\owedge$ operation is called the *(normalized) cautious rule of combination* [9]. Denoting by $w_{1 \owedge 2}$ the weight function corresponding to $m_1 \owedge m_2$, we thus have

$$w_{1 \owedge 2} = w_1 \vee w_2, \tag{13}$$

which is to be compared to (12). The cautious rule is commutative, associative and idempotent, i.e., $m \owedge m = m$ for any nondogmatic mass function m. Also, the orthogonal sum is distributive with respect to $\owedge$: for any nondogmatic mass functions m_1, m_2 and m_3, we have

$$(m_1 \oplus m_2) \owedge (m_1 \oplus m_3) = m_1 \oplus (m_2 \owedge m_3). \tag{14}$$

Eq. (14) lays bare an important property of the cautious rule: when combining two overlapping pieces of evidence ($m_1 \oplus m_2$ and $m_1 \oplus m_3$), the common part (m_1) is not counted twice.

Table 2: The two mass functions of Example 1 and their combination by the cautious rule.

	$\varnothing$	$\{a\}$	$\{b\}$	$\{a,b\}$	$\{c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,b,c\}$
m_1	·	0.5	·	0.3	·	0.1	·	0.1
w_1		0.22	·	1.39	·	0.69	·	
m_2	·	·	·	·	·	0.4	0.2	0.4
w_2		·	·	·	−0.182	0.69	0.41	
$m_{1 \owedge 2}$	·	0.4	0.12	0.24	0.04	0.08	0.04	0.08
$w_{1 \owedge 2}$		0.22	·	1.39	·	0.69	0.41	

This property makes this rule suitable for combining nonindependent pieces of evidence. The cautious rule has shown good performance in many practical applications including classifier combination [34], visual tracking [29], face detection [21], expert opinion combination [24, 6] and cooperative perception in vehicular networks [20].

Example 2. *Let us consider again the two mass functions m_1 and m_2 of Example 1. Table 2 shows these mass functions and their weight assignments, together with the result of their combination by the cautious rule. We observe that $m_{1 \owedge 2} = m_1 \owedge m_2$ is separable. More generally, the combination of an arbitrary mass function with a separable mass function by the cautious rule is always separable.*

It must be noted that the decision to use Dempster's rule or the cautious rule must be based on the consideration of the evidence on which agents have based the construction of their mass functions, Dempster's rule requiring the bodies of evidence to be entirely distinct [36]. Typically, the independence assumption can be considered to hold if agents use distinct sensor data, and it does not hold if they share common information or knowledge. In case of doubt, it may be preferable to use the more conservative cautious rule, which is less likely to yield overconfident conclusions.

3. Distributed orthogonal-sum and cautious combination

In this section, we show how Dempster's rule and the cautious rule can be implemented in a distributed way. We assume that n agents are located at the nodes of a communication graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of vertices, $\mathcal{E}(t) \subseteq \mathcal{V}^2$ is the set of edges and t is a discrete time index (Figure 1). Each agent i holds a mass function m_i and can communicate only with its neighbors in the graph. The communication graph is usually determined by physical constraints such as spatial distance and transmitter power in the case of wireless communication. When agent i can receive information from agent j, it is often the case that agent j can also receive information from agent i, i.e., the communication graph is undirected. While this assumption is not necessary for the subsequent analysis, it is often reasonable and we will adopt it hereafter to simplify the exposition.

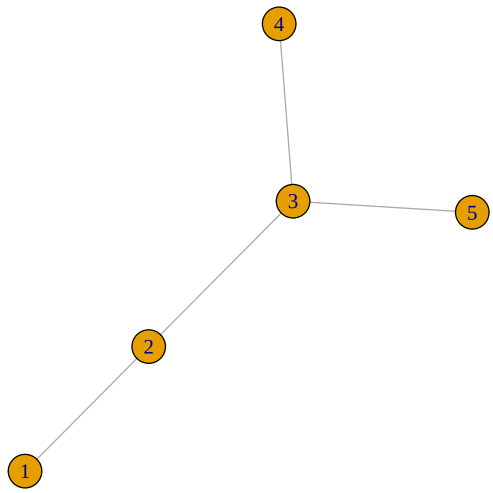


Figure 1: Example of a connected graph with five vertices.

We wish to design *distributed procedures* whereby each agent can combine its mass function with those of other agents by any of the combination rules reviewed in Section 2. As a result, a consensus will be reached, each agent having the same mass function $m_* = m_1 * \ldots * m_n$, with $* = \oplus$ or $* = \owedge$. The key observation is that, as shown in Sections 2.2 and 2.3, combination by Dempster's rule and the cautious rule can be achieved by computing, respectively, the sum and the maximum of the weights of evidence. Consequently, these rules can be implemented in a distributive way using average and maximum consensus algorithms [23, 32].

We note that, in this paper, we consider only the *static consensus* problem in which each agent i first constructs a *fixed* mass function m_i based, e.g., on sensor information in a first step, and then combines it with the mass functions of other agents in the network in a second step. A more difficult problem would be to consider situations in which agents continue to update their beliefs by collecting evidence from the outside world while simultaneously exchanging information with other agents in the network. The combination of dynamically changing reference signals in a decentralized fashion is referred to as the *dynamic consensus* problem [27]. This problem is more complex than the static one; in particular, the convergence of dynamic consensus algorithms can only be guaranteed under some assumptions on the dynamics of the individual reference signals. In practice, static consensus algorithms are applicable and are commonly used when the reference signals (here, individual mass functions) are either fixed, or are updated at a low frequency relative to the convergence time of a static consensus algorithm. This is the case, for instance, in mobile robotics application such as described in [32] or [20].

In the following, static consensus algorithms will first be reviewed in Section 3.1. The application to belief function combination will then be exposed in Section 3.2.

3.1. Consensus algorithms

Let us assume that each agent i in the network initially holds a quantity or "state" $x_i(0)$, and agents update their states at discrete times $t = 1, 2, \ldots$, computing $x_i(t+1)$ from $x_i(t)$ using only the information from their neighbors in the communication graph $\mathcal{G}(t)$. The *consensus problem* is to design an update equation such that the states of all agents converge to the same value ξ, i.e., for all $i \in \{1, \ldots, n\}$,

$$\lim_{t \to \infty} x_i(t) = \xi.$$

The *average* and *maximum consensus* problems correspond, respectively, to the cases where $\xi = \frac{1}{n} \sum_{i=1}^{n} x_i(0)$, and $\xi = \max_{1 \le i \le n} x_i(0)$. In the following, we focus on the average consensus problem first, and we address the simpler maximum consensus problem at the end of this section. Most of the material in this section is drawn from [23] and [32].

Preliminary definitions. Let $\boldsymbol{C} = (c_{ij}) \in \mathbb{R}^{n \times n}$ be an $n \times n$ square matrix. It is said to be *stochastic* if $c_{ij} \ge 0$ for all (i,j), and $\sum_{j=1}^{n} c_{ij} = 1$ for all i, i.e., each row sums to unity. If, additionally, $\sum_{i=1}^{n} c_{ij} = 1$ for all j, i.e., each column also sums to unity, then $\boldsymbol{C}$ is said to be *doubly stochastic*. Obviously, all symmetric stochastic matrices are doubly stochastic. The graph $\mathcal{G}_{\boldsymbol{C}}$ of stochastic matrix $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ is defined as $(\mathcal{V}, \mathcal{E}_{\boldsymbol{C}})$ with $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E}_{\boldsymbol{C}} = \{(i,j) \in \mathcal{V}^2 : c_{ij}(t) > 0\}$. Matrix $\boldsymbol{C}$ is *compatible* with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if its graph $\mathcal{G}_{\boldsymbol{C}} = (\mathcal{V}, \mathcal{E}_{\boldsymbol{C}})$ is a subgraph of $\mathcal{G}$, i.e., $\mathcal{E}_{\boldsymbol{C}} \subseteq \mathcal{E}$. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *undirected* if $(i,j) \in \mathcal{E}$ implies $(j,i) \in \mathcal{E}$. An undirected graph is *connected* if there is a path from any node to any other node in the graph. The set of *neighbors* of vertex $i \in \mathcal{V}$ in undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as $\mathcal{N}(i) = \{j \in \mathcal{V} \setminus \{i\} : (i,j) \in \mathcal{E}\}$. The *union* of two graphs $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2)$ with the same set of vertices is defined as $\mathcal{G}_1 \cup \mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_1 \cup \mathcal{E}_2)$.

Linear consensus algorithm. We consider the following linear update equation

$$\boldsymbol{x}(t+1) = \boldsymbol{C}(t)\boldsymbol{x}(t), \tag{15}$$

where $\boldsymbol{x}(t) = (x_1(t), \ldots, x_n(t))^T$ is the column vector of states at time t and $\boldsymbol{C}(t) \in \mathbb{R}^{n \times n}$ is a stochastic matrix compatible with the communication graph. Eq. (15) can be written as:

$$x_i(t+1) = \sum_{j=1}^{n} c_{ij}(t) x_j(t) = x_i(t) + \sum_{j \in \mathcal{N}(i)} c_{ij}(t)(x_j(t) - x_i(t))$$

for $i = 1, \ldots, n$. Each agent i thus updates its state based on the states on its neighbors in the communication graph.

Several theoretical results guarantee the convergence to the average of the initial quantities $x_i(0)$. Let us first consider the case where the communication graph (assumed to be undirected) remains fixed during the execution of the algorithm and $\boldsymbol{C}(t) = \boldsymbol{C}$ is a constant matrix. Then, the following result holds [23, Theorem 3.1].

Proposition 1. *If $\boldsymbol{C}$ is symmetric, stochastic and such that $c_{ii} > 0$ for all i, and if $\mathcal{G}_{\boldsymbol{C}}$ is connected, then*

$$\lim_{t\to\infty} \boldsymbol{C}^t = \frac{1}{n}\mathbf{1},$$

where $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^n$, t is a positive integer and

$$\boldsymbol{C}^t = \underbrace{\boldsymbol{C}\ldots\boldsymbol{C}}_{t \text{ times}}.$$

Consequently, the linear update equation (15) *solves the average consensus problem.*

As shown by Xiao and Boyd [45], the convergence rate of the linear consensus algorithm (15) with constant weight matrix $\boldsymbol{C}(t) = \boldsymbol{C}$, defined as

$$r(\boldsymbol{C}) = \sup_{\boldsymbol{x}(0)\neq\overline{\boldsymbol{x}}} \lim_{t\to\infty} \left(\frac{\|\boldsymbol{x}(t) - \overline{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}(t) - \overline{\boldsymbol{x}}\|_2} \right)^{1/t},$$

where $\overline{\boldsymbol{x}}$ is the n-vector whose components are the average $(1/n)\sum_{i=1}^n x_i(0)$, is equal to the spectral radius of matrix $\boldsymbol{C} - (1/n)\mathbf{1}\mathbf{1}^T$. A method to design matrix $\boldsymbol{C}$ so as to maximize $r(\boldsymbol{C})$ is described in [45].

If the communication connectivity varies over time, then convergence to the average can still be guaranteed provided the union of the communication graphs $\mathcal{G}(t)$ over a time window of given length is connected. This is expressed by the following proposition [23, Theorem 3.2].

Proposition 2. *Consider a sequence of symmetric stochastic matrices $\{\boldsymbol{C}(t)\}_{t=0}^{+\infty}$ verifying $c_{ii}(t) > 0$ for all i. Let $\mathcal{G}(t)$ denote the graph of $\boldsymbol{C}(t)$. Then the sequence $\boldsymbol{C}(t)$ solves the average consensus problem if and only if there exists a positive integer T such that the graphs $\overline{\mathcal{G}}(t) = \mathcal{G}(t) \cup \mathcal{G}(t+1) \cup \ldots \cup \mathcal{G}(t+T-1)$ for $t = 0, 1, \ldots$ are all connected.*

When the weight matrix varies, the analysis of the convergence rate has to be based on worst-case analysis [23]. Bounds on the rate of convergence subject to constraints on the topological properties of the communication graphs and on the numerical values for the entries of $\boldsymbol{C}(t)$ are given in [1]. Other theoretical results pertain to the case where matrices $\{\boldsymbol{C}(t)\}$ are generated randomly. The reader is referred to [23] for a review of these results.

Design of matrices $\boldsymbol{C}(t)$. To implement a consensus strategy, we need to design either a single matrix $\boldsymbol{C}$ in the case of a fixed communication network, or a sequence of matrices $\boldsymbol{C}(t)$ if the connectivity changes. Optimal design methods, maximizing some global performance criterion, often require a centralized mechanism taking into account the topology of the whole network [45]. Here, we focus on *local* design methods, in which each agent can design its consensus update weights using only information provided by its neighbors.

A common choice for the weight matrix $\boldsymbol{C}$ when the network topology is fixed and agents update their state in a synchronized way is the matrix of Metropolis-Hastings weights [46],

$$c_{ij} = \begin{cases} \frac{1}{\max(d(i),d(j))+1} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j \\ 1 - \sum_{j=1, i\neq j}^n c_{ij} & \text{if } i = j, \end{cases} \tag{16}$$

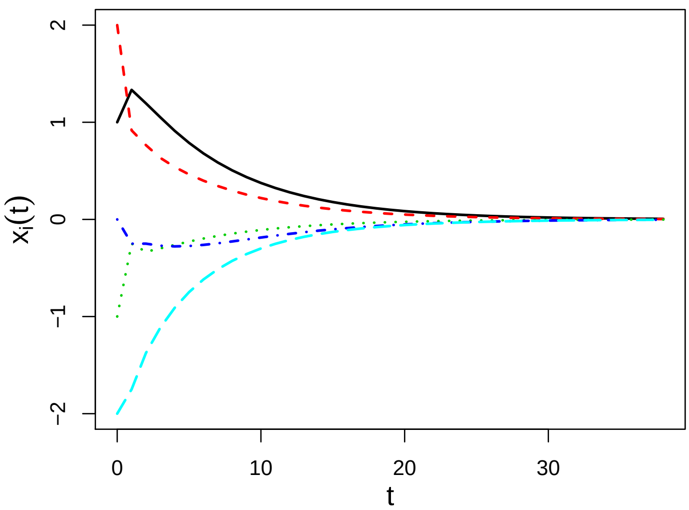


Figure 2: Convergence of states to the average using Metropolis-Hastings weights (Example 3). Each of the five curves represents the state of one of the five nodes as a function of the number of iterations. All states converge to the average $(1 + 2 - 1 + 0 - 2)/5 = 0$ of the initial values.

where $d(i) = |\mathcal{N}(i)|$ is the number of neighbors of node i (called the *degree* of node i). Clearly, this matrix is symmetric, stochastic and its diagonal elements are strictly positive. Furthermore, $\mathcal{G}_{\boldsymbol{C}} = \mathcal{G}$. It thus satisfies the condition of Proposition 1 as long as $\mathcal{G}$ is connected, and it ensures average consensus.

Example 3. *For the graph of Figure 1, the matrix of Metropolis-Hastings weights is*

$$\boldsymbol{C} = \begin{pmatrix} 0.667 & 0.333 & 0 & 0 & 0 \\ 0.333 & 0.417 & 0.25 & 0 & 0 \\ 0 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.75 & 0 \\ 0 & 0 & 0.25 & 0 & 0.75 \end{pmatrix}. \tag{17}$$

Figure 2 shows the convergence of states to the average using the update rule (15) *with this weight matrix, starting from the initial condition* $\boldsymbol{x}(0) = (1, 2, -1, 0, -2)$.

In some real applications such as, e.g., those involving mobile robots exchanging information through an ad hoc network, the assumptions of fixed communication graph and synchronous communication cannot be made. As an example of a consensus strategy allowing for time-varying graphs and asynchronous communication, we can mention the *symmetric gossip* scheme [2, 23], in which at each time step a node i transmits its information to one of its neighbors j, which in turn transmits back its information to i. After this information exchange, both nodes update their state using a consensus scheme. More formally, given a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, assume that at each time t a link $(i,j) \in \mathcal{E}$ with $i \neq j$ is selected, and matrix $\boldsymbol{C}(t)$ is defined as

$$\boldsymbol{C}(t) = \boldsymbol{C}^{ij} = \boldsymbol{I}_n - \frac{1}{2}(\boldsymbol{e}_j - \boldsymbol{e}_i)(\boldsymbol{e}_j - \boldsymbol{e}_i)^T, \tag{18}$$

where $\boldsymbol{I}_n$ is the identity matrix of dimension n, and $\boldsymbol{e}_i$ is the n-vector (i.e., column vector of dimension n) containing only zeros except for the i-th entry, which is set to one. Matrices $\boldsymbol{C}^{ij}$ verify the conditions stated in Proposition 2. If there exists an integer T such that all links are selected at least once in every time window of length T, the joint graphs $\overline{\mathcal{G}}(t) = \bigcup_{i<j} \boldsymbol{C}^{ij}$ are connected, and the sequence of matrices $\boldsymbol{C}(t)$ solves the average consensus problem.

Example 4. *The graph of Figure 1 has four links, with the following corresponding matrices:*

$$\boldsymbol{C}^{12} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{C}^{23} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{C}^{34} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{C}^{35} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \end{pmatrix}.$$

Figure 3 shows the convergence of states to the average using the update rule (15)*, starting from the same initial condition as in Example 3, and defining* $\boldsymbol{C}(t)$ *as one of the above four matrices picked in a random order.*

Maximum consensus. Solving the maximum consensus problem means ensuring that the states $x_i(t)$ of all of the n agents converge to the maximum of the initial states $x_1(0), \ldots, x_n(0)$. This can be achieved by the following nonlinear update rule:

$$x_i(t+1) = \max_{j \in \{i\} \cup \mathcal{N}(i)} x_j(t), \tag{19}$$

i.e., at each time step, each agent updates its state by the maximum of its previous state and the states of its neighbors [42, 32]. This algorithm solves the maximum consensus problem in a finite number of iterations under any of the following two conditions, which mirror the conditions of Propositions 1 and 2:

1. The communication graph $\mathcal{G}$ is fixed and connected.
2. There exists a positive integer T such that the graphs $\overline{\mathcal{G}}(t) = \mathcal{G}(t) \cup \mathcal{G}(t+1) \cup \ldots \cup \mathcal{G}(t+T-1)$ for $t = 0, 1, \ldots$ are all connected.

Example 5. *Coming back to the network of Figure 1, assume that the initial condition is the initial condition* $\boldsymbol{x}(0) = (1, 2, -1, 0, -2)$*. Assuming the five agents update their states in a synchronized way, we have* $\boldsymbol{x}(1) = (2, 2, 2, 0, -1)$ *and* $\boldsymbol{x}(2) = (2, 2, 2, 2, 2)$*, i.e., the procedure converges in only two iterations.*

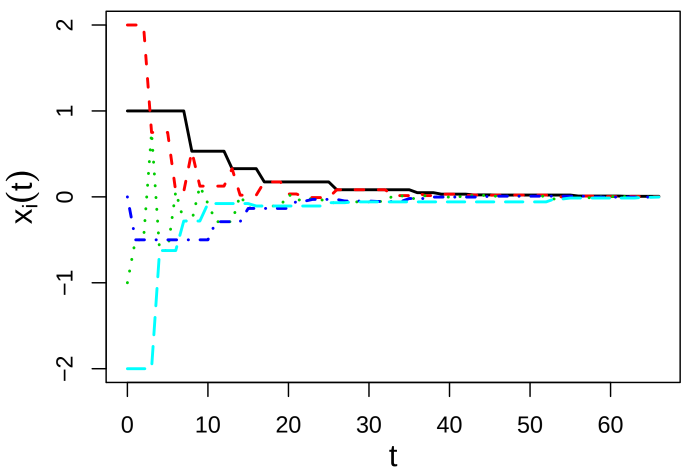


Figure 3: Convergence of states to the average using a symmetric gossip scheme (Example 4). Each of the five curves represents the state of one of the five nodes as a function of the number of iterations. All states converge to the average $(1 + 2 - 1 + 0 - 2)/5 = 0$ of the initial values.

3.2. Application to belief function combination

Let us now assume that agents have collected evidence about some question, and they wish to combine this evidence by exchanging information through the network without any centralized mechanism. To reach this goal, the agents must first agree on the frame of discernment and on a common representation of the available evidence, which can then be combined using the cautious rule or Dempster's rule. These different steps are described below.

Frame of discernment. The first problem to solve is the definition of the frame of discernment Ω. In some applications, this frame is fixed: for instance, in an intrusion detection problem, the frame has only two elements corresponding to the presence or absence of an intruder. In other applications, the frame may potentially be very large, and it may be computationally advantageous to restrict it to as few hypotheses as possible, by discarding hypotheses that are deemed impossible by all agents. For instance, assume that a fleet of mobile robots is searching for a lost object, and the search space is divided into a finite, but large number of cells. If all agents agree that some cells are impossible, then the frame of discernment should include only those cells that are considered possible by at least one agent. Let $\Theta = \{\theta_1, \ldots, \theta_N\}$ be an ordered set containing all hypotheses possible *a priori*, shared by all agents. Initially, each agent i defines a set $\Omega_i(0) \subseteq \Theta$ containing the hypotheses it considers possible, given its own evidence. Then the update rule can be

$$\Omega_i(t+1) = \bigcup_{j \in \{i\} \cup \mathcal{N}(i)} \Omega_j(t) \tag{20}$$

for $j = 1, \dots, n$. This "disjunctive" procedure will converge to $\Omega = \bigcup_{i=1}^n \Omega_i$ in a finite number of iterations. We note that a less conservative approach would be to replace union by intersection in (20), which would amount to discarding any hypothesis considered impossible (or maybe only forgotten or neglected) by even a single agent. This "conjunctive" procedure will then converge to $\bigcap_{i=1}^n \Omega_i$. While this approach may be preferred in some applications in which the agents' opinions are not expected to be conflicting, it is less robust than the previous one, and the disjunctive approach is arguably safer in many applications.

Evidence representation. After agents have agreed on an ordered frame of discernment $\Omega = \{\omega_1, \dots, \omega_K\}$, they can represent their evidence by nondogmatic mass functions m_i on Ω corresponding to weight assignments $w_i : 2^\Omega \setminus \{\varnothing, \Omega\} \to \mathbb{R}$. A simple approach would be to represent each weight assignment by a vector of length $2^K - 2$ but, as mentioned in Section 2.2, most of the weights $w(A)$ will usually be null, except for a small number of supported sets. For large K, agents can limit the amount of communication by first sharing their collections of supported sets, which can be done as follows. Each subset $A \subseteq \Omega$ can be represented by a binary vector $\boldsymbol{a} = (a_1, \dots, a_K) \in \{0,1\}^K$ such that $a_j = 1$ if $\omega_j \in A$ and $a_j = 0$ otherwise. This vector can alternatively be coded as an integer $\alpha = \sum_{j=1}^K a_j 2^{j-1}$. Let $\mathcal{F}_i(0)$ denote the set of integers representing the supported sets of agent i, i.e., the subsets $A \subseteq \Omega$ such that $w_i(A) > 0$. The consensus algorithm based on the following update rule

$$\mathcal{F}_i(t+1) = \bigcup_{j \in \{i\} \cup \mathcal{N}(i)} \mathcal{F}_j(t) \tag{21}$$

converges to $\mathcal{F} = \{\alpha_1, \dots, \alpha_r\} = \bigcup_{i=1}^n \mathcal{F}_i(0)$. Each weight assignment w_i can then be coded as an r-vector $\boldsymbol{w}_{i\cdot}(0) = (w_{i1}(0), \dots, w_{ir}(0))^T$, where $w_{ij}(0)$ is the weight intially assigned by agent i to subset A_j represented by integer α_j.

Example 6. *Consider again five agents at the nodes of the graph of Figure 1. Assume that $\Theta = \{\theta_1, \dots, \theta_{100}\}$. Representing the evidence of each agent by a vector of 2^{100} masses or commonalities would obviously be infeasible. Assume that the initial sets of possibilities of the five agents are*

$$\Omega_1(0) = \{\theta_{10}, \theta_{11}, \theta_{13}, \theta_{14}\}, \quad \Omega_2(0) = \{\theta_{12}, \theta_{13}, \theta_{14}, \theta_{15}\}$$

$$\Omega_3(0) = \{\theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}\}, \quad \Omega_4(0) = \{\theta_{12}, \theta_{13}, \theta_{15}, \theta_{16}\}$$

$$\Omega_5(0) = \{\theta_{11}, \theta_{13}, \theta_{15}, \theta_{16}\}.$$

After a finite number t of iterations, the consensus algorithm defined by (20) converges to

$$\Omega_i(t) = \Omega = \{\theta_{10}, \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}, \theta_{15}, \theta_{16}\}$$

for $i = 1, \dots, 5$. Let us rename the elements of Ω as $\Omega = \{\omega_1, \dots, \omega_7\}$. This is now the common frame of discernment for the five agents. Assume that the agents assess the evidence known to them and represent it by the following separable mass functions:

$$m_1 = \{\omega_4\}^{0.5} \oplus \{\omega_2, \omega_4\}^{0.8}, \quad m_2 = \{\omega_3\}^1 \oplus \{\omega_3, \omega_4\}^{1.2}$$

$$m_3 = \{\omega_3, \omega_4\}^{0.3} \oplus \{\omega_3, \omega_4, \omega_5\}^{0.9}, \quad m_4 = \{\omega_4, \omega_6, \omega_7\}^{0.2}$$

$$m_5 = \{\omega_2, \omega_4\}^{0.7} \oplus \{\omega_2, \omega_4, \omega_6, \omega_7\}^{1.1}.$$

For m_1, the supported sets are $\{\omega_4\}$ and $\{\omega_2, \omega_4\}$. Their binary representations are, respectively, $\boldsymbol{a}_1 = (0,0,0,1,0,0,0)$ and $\boldsymbol{a}_2 = (0,1,0,1,0,0,0)$, and their corresponding integer representations are $\alpha_1 = 2^3 = 8$ and $\alpha_2 = 2^1 + 2^3 = 10$. Consequently, $\mathcal{F}_1(0) = \{8, 10\}$. Similarly, $\mathcal{F}_2(0) = \{4, 12\}$, $\mathcal{F}_3(0) = \{12, 28\}$, $\mathcal{F}_4(0) = \{104\}$ and $\mathcal{F}_5(0) = \{10, 106\}$. The consensus algorithm based on update equation (21) converges after a finite number of iterations to the union of the $\mathcal{F}_i(0)$, i.e.,

$$\mathcal{F} = \{4, 8, 10, 12, 28, 104, 106\}.$$

The weight assignments can, thus, be represented as vectors of length $r = |\mathcal{F}| = 7$ as follows:

	α_j						
	4	8	10	12	28	104	106
$\boldsymbol{w}_{1\cdot}(0)$	0	0.5	0.8	0	0	0	0
$\boldsymbol{w}_{2\cdot}(0)$	1	0	0	1.2	0	0	0
$\boldsymbol{w}_{3\cdot}(0)$	0	0	0	0.3	0.9	0	0
$\boldsymbol{w}_{4\cdot}(0)$	0	0	0	0	0	0.2	0
$\boldsymbol{w}_{5\cdot}(0)$	0	0	0.7	0	0	0	1.1

Cautious combination. As explained in Section 2.3, the cautious rule consists in computing the maximum of weights of evidence. This can be done in a distributed way by applying the update rule (19) to each components of vectors $\boldsymbol{w}_i$ in parallel. More precisely, at each time $t+1$, each agent i updates its weights $w_{ij}(t)$ as

$$w_{ij}(t+1) = \max_{j \in \{i\} \cup \mathcal{N}(i)} w_{ij}(t), \quad j = 1, \ldots, r.$$

Each iteration of this maximum consensus algorithm thus requires rd_i maximum operations at each node i, where $d_i = |\mathcal{N}(i)|$ is the degree of note i. Under the conditions specified in Section 3.1, this algorithm converges in finite number t of iterations to

$$w_{1j}(t) = w_{2j}(t) = \ldots = w_{nj}(t) = w_{\vee j} = \max_i w_{ij}(0), \quad j = 1, \ldots, r.$$

Each agent can then compute the mass function $m_{\owedge} = m_1 \owedge \ldots \owedge m_n$ as

$$m_{\owedge} = \bigoplus_{j=1}^{r} A_j^{w_{\vee j}}, \tag{22}$$

where A_j denotes the subset corresponding to α_j. The number of operations needed to compute the orthogonal sum of r mass functions, each having two focal sets in a frame of size K, is bounded upwards by $K2^{r+1}$, but it is often much less, as the number of focal sets of the final combined mass function is often much less than 2^r [43].

Example 7. *Continuing Example 6, let us assume that the five agents update the weight vectors* $\boldsymbol{x}_i(t)$ *synchronously. The sequence of states is as follows:*

	α_j						
	4	8	10	12	28	104	106
$\boldsymbol{w}_{1\cdot}(1)$	1	0.5	0.8	1.2	0.0	0.0	0.00
$\boldsymbol{w}_{2\cdot}(1)$	1	0.5	0.8	1.2	0.9	0.0	0.0
$\boldsymbol{w}_{3\cdot}(1)$	1	0.0	0.7	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{4\cdot}(1)$	0	0.0	0.0	0.3	0.9	0.2	0.0
$\boldsymbol{w}_{5\cdot}(1)$	0	0.0	0.7	0.3	0.9	0.0	1.1

	α_j						
	4	8	10	12	28	104	106
$\boldsymbol{w}_{1\cdot}(2)$	1	0.5	0.8	1.2	0.9	0.0	0.0
$\boldsymbol{w}_{2\cdot}(2)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{3\cdot}(2)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{4\cdot}(2)$	1	0.0	0.7	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{5\cdot}(2)$	1	0.0	0.7	1.2	0.9	0.2	1.1

	α_j						
	4	8	10	12	28	104	106
$\boldsymbol{w}_{1\cdot}(3)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{2\cdot}(3)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{3\cdot}(3)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{4\cdot}(3)$	1	0.5	0.8	1.2	0.9	0.2	1.1
$\boldsymbol{w}_{5\cdot}(3)$	1	0.5	0.8	1.2	0.9	0.2	1.1

The network thus converges in only three iterations to the weight assignment vector $\boldsymbol{w}_\vee = (1, 0.5, 0.8, 1.2, 0.9, 0.2, 1.1)$*, which corresponds to the following mass function:*

$$m_{\oslash} = \{\omega_3\}^1 \oplus \{\omega_4\}^{0.5} \oplus \{\omega_2,\omega_4\}^{0.8} \oplus \{\omega_3,\omega_4\}^{1.2} \oplus \\ \{\omega_3,\omega_4,\omega_5\}^{0.9} \oplus \{\omega_4,\omega_6,\omega_7\}^{0.2} \oplus \{\omega_2,\omega_4,\omega_6,\omega_7\}^{1.1},$$

equal to

$$m_{\oslash}(\{\omega_3\}) = 0.1132, \quad m_{\oslash}(\{\omega_4\}) = 0.7697, \quad m_{\oslash}(\{\omega_2,\omega_4\}) = 0.0297$$

$$m_{\oslash}(\{\omega_3,\omega_4\}) = 0.0460, \quad m_{\oslash}(\{\omega_3,\omega_4,\omega_5\}) = 0.0118, \quad m_{\oslash}(\{\omega_4,\omega_6,\omega_7\}) = 0.0054$$

$$m_{\oslash}(\{\omega_2,\omega_4,\omega_6,\omega_7\}) = 0.0162, \quad m_{\oslash}(\Omega) = 0.0081.$$

Combination by Dempster's rule. As shown in Section 2.2, Dempster's rule can be implemented by summing the weight assignment vectors. In order to use the linear average consensus algorithm based on update rule (15) for that purpose, agents need to know the total number n of nodes (vertices) in the network. This number can be set in advance in the case of a fixed communication network, but it needs to be estimated in the case of a dynamic time-varying network. Distributed methods for counting the number of nodes in a network are described in [23] and [32, pages 61–62]. We only give the former simpler method here. Assume that the initial conditions are $x_1(0) = 1$ and $x_i = 0$ for $i = 2, \ldots, n$ (i.e., the initial states are 1 for one agent and 0 for the others). If the state variables $x_i(t)$ are updated using the average consensus algorithm, we have

$$\lim_{t \to \infty} x_i(t) = \frac{1}{N} \sum_{i=1}^{n} x_i(0) = \frac{1}{n}.$$

Denoting $\widehat{n}_i(t) = 1/x_i(t)$, we thus have $\lim_{t\to\infty} \widehat{n}_i(t) = n$.

Once the number of nodes has been estimated, the algorithm defined by update equation (15) can be used to compute the average weights of evidence assigned to each supported set in a distributed way. After convergence, the average weights will be multiplied by n to obtain the sum of the weights. Let $\boldsymbol{w}_{\cdot j}(0) = (w_{1j}(0), \ldots, w_{nj}(0))^T$ be the n-vector of weights of evidence for supported set A_j initially held by the n agents. The partial state vectors $\boldsymbol{w}_{\cdot j}(t)$ for $j = 1, \ldots, r$ can be updated in parallel at time $t+1$ using linear equations

$$\boldsymbol{w}_{\cdot j}(t+1) = \boldsymbol{C}\boldsymbol{w}_{\cdot j}(t). \tag{23}$$

Let $\underline{\boldsymbol{w}}(t) = (\boldsymbol{w}_{\cdot 1}(t)^T, \ldots, \boldsymbol{w}_{\cdot r}(t)^T)^T$ be the rn-vector obtained by concatenating the r vectors $\boldsymbol{w}_{\cdot j}(t)$, $j = 1, \ldots, r$, and let $\underline{\boldsymbol{C}}$ be the $rn \times rn$ matrix $\underline{\boldsymbol{C}} = \boldsymbol{I}_r \otimes \boldsymbol{C}$, where $\boldsymbol{I}_r$ is the identity matrix of size r and $\otimes$ is the Kronecker product. Then, the r update equations (23) can be written more compactly as

$$\underline{\boldsymbol{w}}(t+1) = \underline{\boldsymbol{C}}\,\underline{\boldsymbol{w}}(t). \tag{24}$$

Under the conditions of Propositions 1 or 2, we have

$$\lim_{t\to\infty} n w_{ij}(t) = \sum_{i=1}^{n} w_{ij}(0),$$

i.e., each agent i holds after convergence the r-vector of sums $\boldsymbol{w}_\Sigma = (\sum_{i=1}^n w_{i1}, \ldots, \sum_{i=1}^n w_{ir})$, which makes it possible to compute the orthogonal sum

$$m_\oplus = m_1 \oplus \ldots \oplus m_n = \bigoplus_{j=1}^{r} A_j^{w_{\Sigma j}}. \tag{25}$$

The amount of computation of the distributed Dempster's rule is similar to that of the distributed cautious rule. During the first run of the average consensus to compute the number of nodes, the amount of arithmetic operations at each node i during each iteration is proportional to d_i. During the second run to sum up the weights, it is proportional to rd_i. The computation of $m_\oplus$ from the weight vector $\boldsymbol{w}_\Sigma$ requires $O(K2^{r+1})$ arithmetic operations in the worst case.

Example 8. *Considering again the data of Example 6, Figure 4 shows the convergence of quantities $nw_{ij}(t)$ to $\sum_{i=1}^n w_{ij}(0)$, using the Metropolis-Hastings matrix* (17) *in the update*

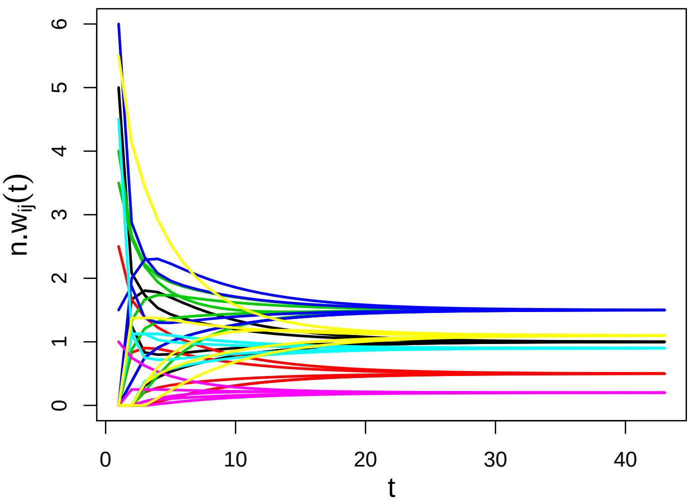


Figure 4: Distributed implementation of Dempster's rule: convergence of $nw_{ij}(t)$ to the sum $\sum_{i=1}^{n} w_{ij}(0)$ of initial weights (Example 8). There are $n = 5$ agents and $r = 7$ supported sets A_j, which makes $5 \times 7 = 35$ curves. We used separate color for each supported set. The values at convergence are $\boldsymbol{w}_{\Sigma} = (1.0, 0.5, 1.5, 1.5, 0.9, 0.2, 1.1)$.

equations (23). *The network converges to* $\boldsymbol{w}_{\Sigma} = (1.0, 0.5, 1.5, 1.5, 0.9, 0.2, 1.1)$, *corresponding to the orthogonal sum*

$$m_{\oplus} = \{\omega_3\}^1 \oplus \{\omega_4\}^{0.5} \oplus \{\omega_2, \omega_4\}^{1.5} \oplus \{\omega_3, \omega_4\}^{1.5} \oplus \\ \{\omega_3, \omega_4, \omega_5\}^{0.9} \oplus \{\omega_4, \omega_6, \omega_7\}^{0.2} \oplus \{\omega_2, \omega_4, \omega_6, \omega_7\}^{1.1},$$

equal to

$$m_{\oplus}(\{\omega_3\}) = 0.0596, \quad m_{\oplus}(\{\omega_4\}) = 0.8644, \quad m_{\oplus}(\{\omega_2, \omega_4\}) = 0.0329$$
$$m_{\oplus}(\{\omega_3, \omega_4\}) = 0.0269, \quad m_{\oplus}(\{\omega_3, \omega_4, \omega_5\}) = 0.0046, \quad m_{\oplus}(\{\omega_4, \omega_6, \omega_7\}) = 0.0021$$
$$m_{\oplus}(\{\omega_2, \omega_4, \omega_6, \omega_7\}) = 0.0063, \quad m_{\oplus}(\Omega) = 0.0031.$$

4. Robust combination

Both Dempster's rule and the cautious rule assume the combined pieces of evidence to be reliable. For this condition to hold, it is necessary, in particular, that the different agents address the same question. There are, however, situations in which this assumption cannot be guaranteed to be verified. Consider, for instance, the data association problem [12, 30], in which agents collect sensor information about a set of objects. Before this information can be combined, we need to match the objects perceived by each pair of agents. If matching errors occur, then the beliefs held by one agent about some object may be erroneously combined with the beliefs held by the other agent about another object. If the number of agents is large enough and if the matching error probability is small, we can expect to observe consensus among a majority of agents and a few outliers. A robust combination procedure should be able to exclude those outliers and combine only information from consistent sources. In this section, we describe such a procedure based on the RANSAC algorithm [22]. The idea of applying a RANSAC-like algorithm to robust combination of belief functions was first proposed by Zair et al. [49, 48]. Our proposal is closer to the original RANSAC algorithm and lends itself to distributed implementation using some ideas from [32, Chapter 4].

In the following, we first recall the RANSAC algorithm and discuss its application to belief function combination in Section 4.1. The distributed implementation is then described in Section 4.2 and an illustrative example is presented in Section 4.3.

4.1. RANSAC algorithm and application to evidence combination

The random sample consensus (RANSAC) algorithm [22] aims to estimate the parameters of a model from a dataset containing a limited number of outliers. There exist several variants of this algorithm, but the method basically consists in randomly selecting a subsample of the dataset and using it to fit a model. All data items then "vote" for that model if they are well explained by it (i.e., if the model error is less than some threshold). The process is repeated a given number of times, and the model with the largest number of votes is selected. Finally, a better model is fitted by including in the subsample all the observations that voted for the winning model.

The algorithm has three parameters: the size ν of each random subsample, the error threshold τ used in the voting process, and the number N of generated models. In applications such as least-squares estimation, ν is usually chosen as the minimum number of observations to fit a model (i.e., $\nu = 2$ for a straight line). The threshold parameter τ is application-specific and reflects how well the user expects a correct model to explain "inliers". To fix N, Fischler and Bolles [22] propose the following line of reasoning. Assume that each observation has a probability p_{in} to be an inlier. The probability that a subsample of size ν is composed of only inliers is p_{in}^ν and, consequently, the probability that a sample of size ν contains at least one outlier is $1 - p_{\text{in}}^\nu$. If N samples are generated, the probability that at least one of them contains only inliers is, thus, $1 - (1 - p_{in}^\nu)^N$. For this probability to be larger that some predefined value p_{suc}, N has to verify the following inequality:

$$N \geq \frac{\log(1 - p_{\text{suc}})}{\log(1 - p_{\text{in}}^\nu)}. \tag{26}$$

When applying the RANSAC approach to evidence combination, the dataset is a set $\mathcal{M} = \{m_1, \ldots, m_n\}$ of n mass functions. Given a subset of indices $\mathcal{I} \subset \{1, \ldots, n\}$ of cardinality ν, a model is the mass function $m_{\mathcal{I}}$ obtained by combining the mass functions $\{m_i : i \in \mathcal{I}\}$. Assuming, as in [49, 48], the mass functions to be combined by Dempster's rule, we have

$$m_{\mathcal{I}} = \bigoplus_{i \in \mathcal{I}} m_i \tag{27}$$

As a measure of discrepancy between a model $m_{\mathcal{I}}$ and a mass function m_j, we can use the degree of conflict between m_j and $m_{\mathcal{I}}$ as

$$\kappa(m_{\mathcal{I}}, m_j) = (m_{\mathcal{I}} \cap\!\!\!\!\bigcirc\, m_j)(\varnothing). \tag{28}$$

In [49, 48], the authors do not select a random subset $\mathcal{I}$ as in the original RANSAC algorithm, but they search the space $2^{\mathcal{M}}$ of all models, adding or removing one mass function at a time. This approach is interesting but it is difficult to implement in a distributed way. For this reason, we stick to the standard RANSAC algorithm in this paper. The centralized (non distributed) procedure is described in Algorithm 1.

Remark 1. *When n is small, the random sampling can be replaced by an exhaustive enumeration of all subsets $\mathcal{I}$ of $\{1, \ldots, n\}$ of cardinality ν.*

Remark 2. *The same approach can be used with the cautious rule instead of Dempster's rule. In that case,* (27) *is replaced by*

$$m_{\mathcal{I}} = \bigcirc\!\!\!\!\!\wedge_{i \in \mathcal{I}}\, m_i \tag{29}$$

and the degree of conflict between $m_{\mathcal{I}}$ and m_j is computed as

$$\kappa_{\bigcirc\!\!\!\!\wedge}(m_{\mathcal{I}}, m_j) = (m_{\mathcal{I}} \wedge\!\!\!\!\cdot\;\, m_j)(\varnothing),$$

where $\wedge\!\!\!\!\cdot\;$ denotes the unnormalized cautious rule [9].

Example 9. *Figure 5 shows $n = 40$ mass functions on $\Omega = \{\omega_1, \omega_2\}$ in barycentric coordinates. The bottom left, bottom right and top vertices correspond, respectively, to focal sets $\{\omega_1\}$, $\{\omega_2\}$ and Ω. Thirty "inlier" mass functions support $\{\omega_1\}$ while 10 outliers (in the lower-left corner) support $\{\omega_2\}$. Algorithm 1 was run with $\tau = 0.3$ and $\nu = 5$. Choosing $p_{suc} = 0.9999$ and $p_{in} = 0.75$ in* (26) *gave us $N = 34$. Figures 5a and 5b show two subsamples $\mathcal{I}$ (marked by crosses), with the corresponding sets of outliers (filled circles) and inliers (unfilled circles). The solution of Figure 5a has a score of 30 and is the optimal solution found by the algorithm. In contrast, Figure 5b shows a suboptimal solution with a score of 6. The intermediate mass functions $m_{\mathcal{I}}$ and the final mass functions m are shown, respectively, as squares and triangles.*

4.2. Distributed implementation

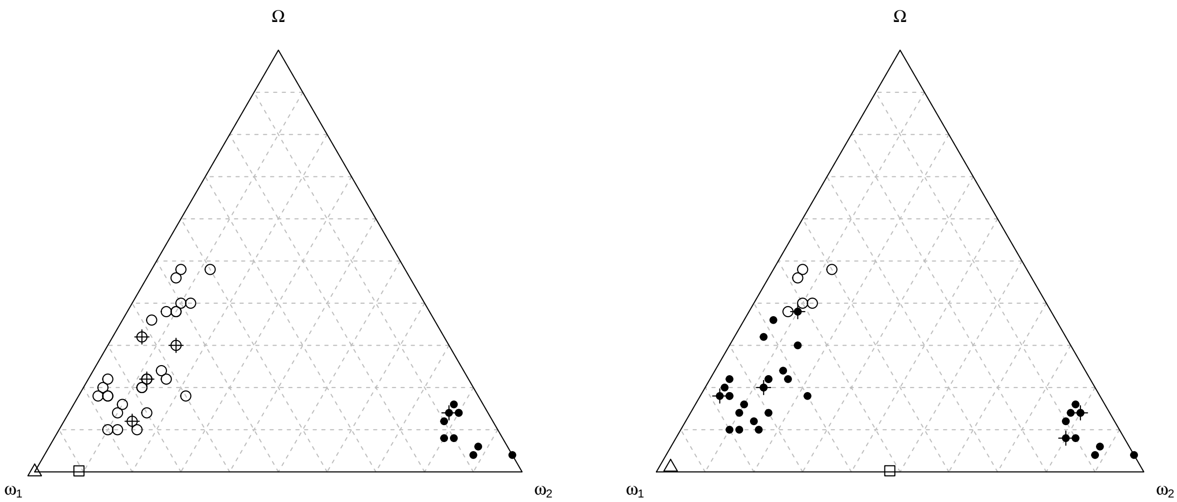
The RANSAC procedure described above can be implemented in a distributed way using ideas introduced in [32, Chapter 4]. As in Section 3.2, we assume n agents to be located at the nodes of a network with fixed or time-varying connectivity and verifying the conditions of Propositions 1 or 2, respectively. We assume that the agents have already agreed on the frame of discernement and the list of supported sets using the distributed procedure described in Section 3.2, and that each agent j holds a weight assignment vector $\boldsymbol{w}_j$ representing some piece of evidence. We wish to design a robust distributed procedure for combining this evidence using either Dempster's rule or the cautious rule. Dempster's rule will be assumed in the rest of this section for ease of exposition. To this end, the following three steps have to be performed without any centralized mechanism: (1) generation of models from random subsamples, (2) voting process and (3) combination of inliers. These steps are described below.

Algorithm 1 Centralized RANSAC algorithm for combining belief functions by Dempster's rule.

Require: Set of mass functions $\mathcal{M} = \{m_1, \dots, m_n\}$, subsample size ν, conflict threshold τ, sucess probability p_{suc}, inlier probability p_{in}

$N \leftarrow \lceil \frac{\log(1-p_{\text{suc}})}{\log(1-p_{\text{in}}^{\nu})} \rceil$
best.score $\leftarrow 0$
for $i = 1$ **to** N **do**
  Draw a set $\mathcal{I}$ of ν elements from $\{1, \dots, n\}$
  score $\leftarrow 0$
  Maybe.Inliers $\leftarrow \varnothing$
  $m_{\mathcal{I}} \leftarrow \bigoplus_{i \in \mathcal{I}} m_i$
  for $j = 1$ **to** n **do**
    $\kappa(m_{\mathcal{I}}, m_j) \leftarrow (m_{\mathcal{I}} \cap m_j)(\varnothing)$
    if $\kappa(m_{\mathcal{I}}, m_j) \leq \tau$ **then**
      score $\leftarrow$ score $+ 1$
      Maybe.Inliers $\leftarrow$ Maybe.Inliers $\cup \{j\}$
    end if
  end for
  if score > best.score **then**
    best.score $\leftarrow$ score
    Inliers $\leftarrow$ Maybe.Inliers
  end if
end for
$m \leftarrow \bigoplus_{i \in \text{Inliers}} m_i$

Ensure: m, Inliers



(a) (b)

Figure 5: Mass functions of Example 9 in barycentric coordinates, for two different subsamples $\mathcal{I}$ of size $\nu = 5$ shown as crosses. Outliers and inliers are shown, respectively, as filled and unfilled circles. The intermediate mass functions $m_{\mathcal{I}}$ and the final mass functions m are shown, respectively, as squares and triangles. The solution on the left-hand side (a) is optimal and has a score of 30. It is the final result of the algorithm. The solution on the right-hand side (b) is suboptimal and has a score of 6.

Model generation. This step can be carried out using a random number generation procedure and the maximum consensus algorithm. Initially, each agent i generates a random number β_i. Let

$$\beta_{(1)} > \beta_{(2)} > \ldots > \beta_{(n)}$$

be the n numbers β_i (assumed to be distinct) in decreasing order. We need a distributed mechanism for computing the ν largest elements $\beta_{(1)}, \ldots, \beta_{(\nu)}$. This can be achieved by defining initial sets $\boldsymbol{\beta}_j(0) = \{\beta_j\}$ and using the following update equation:

$$\boldsymbol{\beta}_j(t+1) = \max{}_\nu \left[\boldsymbol{\beta}_j(t) \cup \left(\bigcup_{k \in \mathcal{N}(j)} \boldsymbol{\beta}_k(t) \right) \right], \tag{30}$$

where $\max_\nu$ selects the ν largest in a set. By iteratively applying (30), each $\boldsymbol{\beta}_j(t)$ converges to the set $\boldsymbol{\beta} = \{\beta_{(1)}, \ldots, \beta_{(\nu)}\}$, and the subsample $\mathcal{I}$ is defined as $\mathcal{I} = \{j : \beta_j \in \boldsymbol{\beta}\}$. Each agent knows whether it belongs to $\mathcal{I}$ or not. Let

$$\boldsymbol{w}_j(0) = \begin{cases} \boldsymbol{w}_j & \text{if } j \in \mathcal{I}, \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

where $\mathbf{0} = (0, \ldots, 0)$ is a vector of zeros, which is the weight assignment vector of the vacuous mass function $m_?$. The average consensus algorithm described in Section 3.2 converges to the average $\frac{1}{n} \sum_{j \in \mathcal{I}} \boldsymbol{w}_j$, which after multiplication by n gives us the weight assignment vector $\boldsymbol{w}_\mathcal{I} = \sum_{j \in \mathcal{I}} \boldsymbol{w}_j$ corresponding to $m_\mathcal{I} = \bigoplus_{j \in \mathcal{I}} m_j$. The process can be executed in parallel to generate N subsamples $\mathcal{I}_1, \ldots, \mathcal{I}_N$ and the corresponding combined weight vectors $\boldsymbol{w}_{\mathcal{I}_1}, \ldots, \boldsymbol{w}_{\mathcal{I}_N}$.

Voting. After the N models have been generated, each agent j votes for them by computing the mass functions $m_{\mathcal{I}_i}$ and the degrees of conflict $\kappa(m_{\mathcal{I}_i}, m_j)$. The votes of agent j may be encoded in the binary N-vector $\boldsymbol{v}_j(0) = (v_{j1}(0), \ldots, v_{jN}(0))$ defined as

$$v_{ji}(0) = \begin{cases} 1 & \text{if } \kappa(m_{\mathcal{I}_i}, m_j) \leq \tau \\ 0 & \text{otherwise.} \end{cases}$$

The vote vectors are then updated using the average consensus algorithm, which after convergence allows each agent to compute the score of each of the N models:

$$\mathsf{score}_i = \sum_{j=1}^{n} v_{ji}.$$

The winning model is the one with the largest of votes and, in case of ties, the smallest index:

$$i^* = \min \left(\arg \max_{1 \leq i \leq N} \mathsf{score}_i \right).$$

Combination of the inliers. The final step of the RANSAC algorithm is the combination of the inlier mass functions. Each agent knows whether it voted for model i^* and thus belongs to the inlier set:

$$\mathsf{Inliers} = \{j \in \{1, \ldots, n\} : v_{ji^*} = 1\}.$$

To combine the mass functions m_j for $j \in \mathsf{Inliers}$, we initialize the weight assignment vectors as

$$\boldsymbol{w}_j(0) = \begin{cases} \boldsymbol{w}_j & \text{if } j \in \mathsf{Inliers}, \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

and execute one more time the average consensus algorithm. After convergence and multiplication by n, each agent gets the weight vector $\boldsymbol{w} = \sum_{j \in \mathsf{Inliers}} \boldsymbol{w}_j$, from which the mass function m can be computed as

$$m = \bigoplus_{k=1}^{r} A_k^{w_k},$$

where $A_1, \ldots, A_r$ are the r supported sets identified at the beginning of the process.

4.3. Illustrative example

As an example, we consider an idealized distributed classification problem in which each agent observes a feature vector whose distribution depends on the class variable $Y \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$. Let X_i be the feature vector observed by agent i and let f_{ik} be its conditional probability density given $Y = \omega_k$. After observing $X_i = x_i$, agent i can use the Generalized Bayesian Theorem (GBT) [39, 14] to derive a mass function m_i on Ω defined as

$$m_i = \bigoplus_{k=1}^{K} (\overline{\{\omega_k\}})^{-\ln \gamma_i f_{ik}(x_i)}, \tag{31}$$

where γ_i is a constant such that

$$\gamma_i \leq [\max_k \sup_x f_{ik}(x)]^{-1},$$

which ensures that $\gamma_i f_{ik}(x_i) \leq 1$ for all k and all x_i. The meaning of (31) is clear: a small value of the density $f_{ik}(x_i)$ is evidence *against* ω_k or, equivalently, *for* the complement $\overline{\{\omega_k\}}$ of ω_k. The weight assignment function corresponding to m_i is, thus,

$$w_i(A) = \begin{cases} -\ln \gamma_i f_{ik}(x_i) & \text{if } A = \overline{\{\omega_k\}} \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

It can be represented by a K-vector $\boldsymbol{w}_i = -(\ln f_{i1}(x_i), \ldots, \ln f_{iK}(x_i)) - \ln \gamma_i$.

Now, assume that the features describe some object but, because of incorrect data association for instance, a minority of agents actually observe a different object belonging to a different class. In that case, the agents' mass functions will be highly conflicting and a robust combination rule must be able to identify and discard outliers.

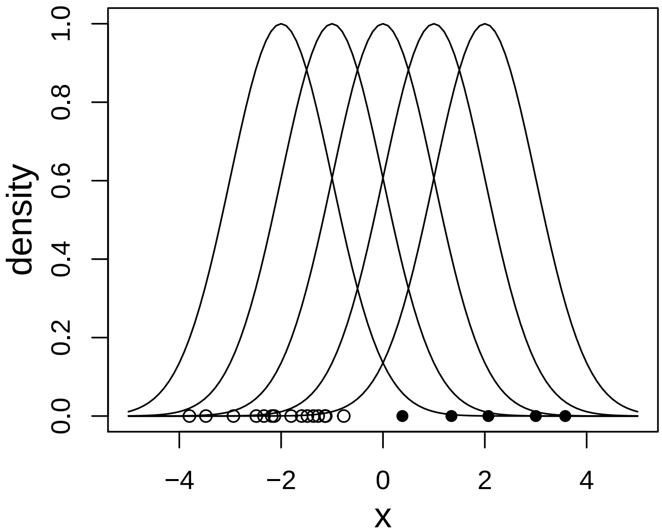
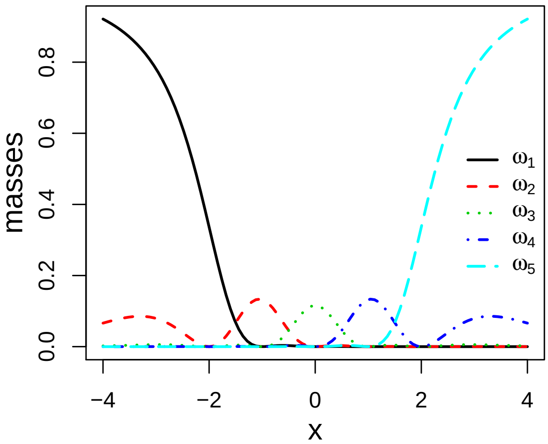
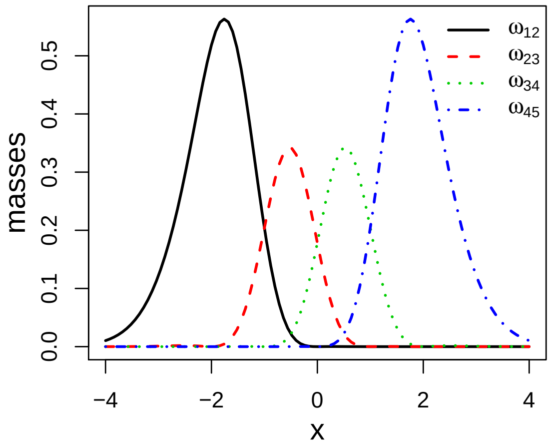


Figure 6: Normal conditional densities of features X_i for the five classes, and example of a dataset. Fifteen observations are drawn from $N(-2,1)$ (unfilled circles) and five are drawn from $N(2,1)$ (filled circles).

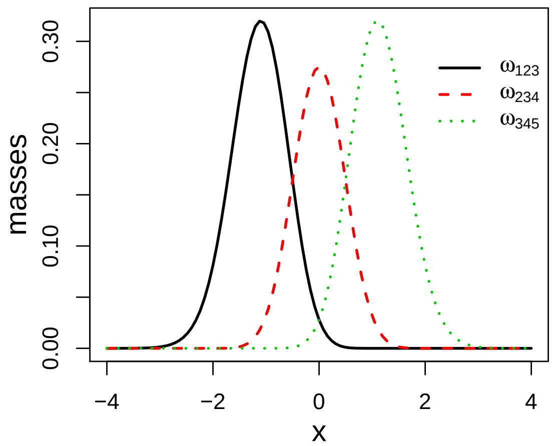
Numerical simulations. We performed simulations with $K = 5$ classes and $n = 20$ agents. Without loss of generality, we assumed that each agent i observes a scalar feature X_i whose conditional distribution given class ω_k was assumed to be normal with mean $\mu_k \in \{-2,-1,0,1,2\}$ and standard deviation $\sigma = 1$ (Figure 6). We considered a scenario in which the object actually belongs to class ω_1 and 15 agents observe a feature x_i drawn for the normal distribution with $\mu_1 = -2$, whereas five agents observe the feature of a class ω_5 object drawn from a normal distribution with mean $\mu_5 = +2$. An example of a dataset $x_1,\ldots,x_{20}$ is shown in Figure 6. The mass function m_i obtained by the GBT (31) is illustrated in Figure 7. The mass assigned to the singletons as well as some of the masses assigned to pairs and triples are shown, respectively, in Figures 7a, 7b and 7c, as functions of x. We applied the RANSAC algorithm to 100 datasets. The number N of samples was computed from (26) with $p_{\text{suc}} = 0.9999$ and $p_{\text{in}} = 0.75$ in (26), giving us $N = 34$. We considered values of $\nu \in \{3,5,10\}$ and $\tau \in \{0.2,0.3,\ldots,0.8\}$. Table 3 shows the false positive rate (proportion of inliers wrongly labeled as outliers, Table 3a) and the false negative rate (proportion of outliers wrongly labeled as inliers, Table 3b) in percent, averaged over the 100 datasets. Parameter ν seems to have a negligible influence on the performance of the procedure. Larger valued of τ induce smaller false positive rates but larger false negative rates. However, the procedure is not too sensitive to τ and values in the range $[0.4,0.6]$ appear to yield good results.



(a)



(b)



(c)

Figure 7: Mass function obtained by the GBT vs. feature value x: masses assigned to the singletons $\{\omega_k\}$, $1 \leq k \leq 5$ (a), to the pairs $\{\omega_k, \omega_{k+1}\}$ for $1 \leq k \leq 4$ (b) and to the triples $\{\omega_k, \omega_{k+1}, \omega_{k+2}\}$ for $1 \leq k \leq 3$ (c). The notations ω_{kl} and ω_{klp} stand, respectively, for $\{\omega_k, \omega_l\}$ and $\{\omega_k, \omega_l, \omega_p\}$. This figure is better viewed in color.

Table 3: False positive rates (a) and false negative rates (b) in % of the RANSAC algorithm with different values of parameters ν and τ.

(a) False positive rates

		τ						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
	3	20.2	11.6	6.67	3.20	1.53	0.87	0.93
ν	5	23.5	14.5	8.20	3.67	1.67	0.67	0.60
	10	24.6	15.6	8.27	4.07	1.87	0.80	0.80

(b) False negative rates

		τ						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
	3	0	0.4	1.0	1.8	4.0	9.2	18.0
ν	5	0	0.2	0.4	1.8	3.4	7.0	14.8
	10	0	0.2	0.6	1.4	3.4	7.6	14.40

5. Conclusions

The DS theory of belief functions has been widely used for information fusion, due to its ability to represent and combine elementary pieces of information such as provided by sensors or expert judgements. A large body of work has been devoted to the development of procedures making it possible to combine belief functions in a wide range of settings, including conflicting or dependent evidence. Most of these procedures, however, assume that all the belief functions to be combined are available at a single central location. This assumption is not verified in applications such as data fusion in sensor networks or collaborative perception in multi-robot systems. In these applications, information is local, each agent holding only a piece of the available evidence representable by a belief function, and agents can only exchange information with their neighbors in a communication network. We then need distributed fusion procedures allowing the combination of belief functions without any centralized mechanism.

Such procedures have been proposed in this paper, based on the notion of weight of evidence. An important observation, already made by Shafer in [35] but often overlooked, is that evidence can usually be broken down into elementary pieces, each one supporting a single hypothesis to some degree. A mass function can, thus, be represented as a collection of subsets of the frame of discernment, with associated weights of evidence. The weight representation has several advantages: it is sparse (most of the propositions usually have zero weight) and it lends itself to easy calculation, as Dempster's rule boils down to addition in weight space. The cautious rule, an idempotent combination operator suitable for combining dependent evidence, replaces summation by the maximum operation. Adopting the weight representation thus turns the distributed combination problem into the computation of the sum or the maximum of some quantities in a network, which can be solved using existing average and maximum consensus algorithms. Although the amount of communication needed to perform these operations is, in the worst case, exponential in the size of the frame of discernement, we have shown that it is usually much less thanks to the sparsity of the weight representation, and we have described distributed procedures allowing agents to agree on the smallest possible frame and list of supported hypotheses. We have also demonstrated the feasibility of a robust combination procedure based on a distributed implementation of the RANSAC algorithm introduced in [32].

Although we have focussed on conjunctive combination in this paper, as this is what is mostly needed in applications, the same principles could be applied to disjunctive combination, using the notion of disjunctive weights introduced in [9]. Another direction of research is to consider situations in which the agent's beliefs vary in time as they continue to collect sensor information while they exchange information in the communication network. To model such situations, we may need to use dynamic consensus algorithms such as reviewed in [27]. Even more fundamentally, we have only considered the simple situation in which all agents hold evidence pertaining to the same question. While this situation is common in many applications, an interesting extension of this work would be to consider the more complex case in which agents hold evidence about different variables and share common knowledge about the relations between these variables. This knowledge could be assumed to be formalized as a valuation-based system (VBS) [37]. We would then have to combine distributed computation in the communication network with local computation in the VBS hypergraph. This challenging problem is left for further research.

Acknowledgment

This research was supported by the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" by the National Agency for Research (reference ANR-11-IDEX-0004-02).

References

[1] D. Angeli and P.-A. Bliman. Tight estimates for non-stationary consensus with fixed underlying spanning tree. *IFAC Proceedings Volumes*, 41(2):9021–9026, 2008. 17th IFAC World Congress.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[3] A. Bronevich and I. Rozenberg. The choice of generalized Dempster-Shafer rules for aggregating belief functions. *International Journal of Approximate Reasoning*, 56:122–136, 2015.

[4] A. G. Bronevich and I. N. Rozenberg. The contradiction between belief functions: Its description, measurement, and correction based on generalized credal sets. *International Journal of Approximate Reasoning*, 112:119–139, 2019.

[5] M. E. G. V. Cattaneo. Belief functions combination without the assumption of independence of the information sources. *International Journal of Approximate Reasoning*, 52(3):299–315, 2011.

[6] Z. L. Cherfi, L. Oukhellou, E. Côme, T. Denœux, and P. Aknin. Partially supervised independent factor analysis using soft labels elicited from multiple experts: Application to railway track circuit diagnosis. *Soft Computing*, 16(5):741–754, 2012.

[7] Y. Dan, J. Hongbing, and G. Yongchan. A robust D-S fusion algorithm for multi-target multi-sensor with higher reliability. *Information Fusion*, 47:32–44, 2019.

[8] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[9] T. Denœux. Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence. *Artificial Intelligence*, 172:234–264, 2008.

[10] T. Denoeux. 40 years of Dempster-Shafer theory. *International Journal of Approximate Reasoning*, 79:1–6, 2016.

[11] T. Denœux, D. Dubois, and H. Prade. Representations of uncertainty in artificial intelligence: Beyond probability and possibility. In P. Marquis, O. Papini, and H. Prade, editors, *A Guided Tour of Artificial Intelligence Research*, volume 1, chapter 4, pages 119–150. Springer Verlag, 2020.

[12] T. Denœux, N. El Zoghby, V. Cherfaoui, and A. Jouglet. Optimal object association in the Dempster-Shafer framework. *IEEE Transactions on Cybernetics*, 44(22):2521–2531, 2014.

[13] T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta. A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning. *International Journal of Approximate Reasoning*, 113:287–302, 2019.

[14] T. Denœux and P. Smets. Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B*, 36(6):1395–1406, 2006.

[15] T. Denœux. Calibrated model-based evidential clustering using bootstrapping. *Information Sciences*, 528:17–45, 2020.

[16] S. Destercke and D. Dubois. Idempotent conjunctive combination of belief functions: Extending the minimum rule of possibility theory. *Information Sciences*, 181(18):3925–3945, 2011.

[17] D. Dubois, F. Faux, and H. Prade. Prejudice in uncertain information merging: Pushing the fusion paradigm of evidence theory further. *International Journal of Approximate Reasoning*, 121:1–22, 2020.

[18] D. Dubois and H. Prade. A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12(3):193–226, 1986.

[19] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4:244–264, 1988.

[20] N. El Zoghby, V. Cherfaoui, and T. Denoeux. Evidential distributed dynamic map for cooperative perception in VANets. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1421–1426, June 2014.

[21] F. Faux and F. Luthon. Theory of evidence for face detection and tracking. *International Journal of Approximate Reasoning*, 53(5):728–746, 2012.

[22] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381?395, June 1981.

[23] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. In A. Bemporad, M. Heemels, and M. Johansson, editors, *Networked Control Systems*, volume LNCIS 406, pages 75–107. Springer-Verlag, Berlin, 2010.

[24] M. Ha-Duong. Hierarchical fusion of expert opinions in the Transferable Belief Model. *International Journal of Approximate Reasoning*, 49(3):555–574, 2008.

[25] D. L. Hall, C.-Y. Chong, J. Llinas, and M. Liggins II, editors. *Distributed Data Fusion for Network-Centric Operations*. CRC Press, Boca Raton, FL, 2013.

[26] O. Kanjanatarakul and T. Denœux. Distributed data fusion in the Dempster-Shafer framework. In *2017 12th System of Systems Engineering Conference (SoSE)*, pages 1–6, Waikoloa, HI, USA, June 2017.

[27] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[28] S. W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus. Multi-vehicle cooperative driving using cooperative perception: Design and experimental validation. *IEEE*

Transactions on Intelligent Transportation Systems, 16(2):663–680, 2015.

[29] J. Klein, C. Lecomte, and P. Miché. Hierarchical and conditional combination of belief functions induced by visual tracking. *International Journal of Approximate Reasoning*, 51(4):410–428, 2010.

[30] H. Laghmara, T. Laurain, C. Cudel, and J. Lauffenburger. Heterogeneous sensor data fusion for multiple object association using belief functions. *Information Fusion*, 57:44–58, 2020.

[31] E. Lefèvre, O. Colot, and P. Vannoorenberghe. Belief function combination and conflict management. *Information Fusion*, 3(2):149–162, 2002.

[32] E. Montijano and C. Sagüés. *Distributed Consensus with Visual Perception in Multi-Robot Systems.* Springer, Heidelberg, 2015.

[33] F. Pichon, D. Dubois, and T. Denœux. Quality of information sources in information fusion. In É. Bossé and G. L. Rogova, editors, *Information Quality in Information Fusion and Decision Making*, pages 31–49. Springer International Publishing, Cham, 2019.

[34] B. Quost, M.-H. Masson, and T. Denœux. Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules. *International Journal of Approximate Reasoning*, 52(3):353–374, 2011.

[35] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, Princeton, N.J., 1976.

[36] G. Shafer. Constructive probability. *Synthese*, 48(1):1–60, 1981.

[37] P. P. Shenoy. A valuation-based language for expert systems. *International Journal of Approximate Reasoning*, 3:383–411, 1989.

[38] P. Smets. The combination of evidence in the Transferable Belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458, 1990.

[39] P. Smets. Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35, 1993.

[40] P. Smets. The canonical decomposition of a weighted belief. In *Int. Joint Conf. on Artificial Intelligence*, pages 1896–1901, San Mateo, Ca, 1995. Morgan Kaufman.

[41] P. Smets. Analyzing the combination of conflicting belief functions. *Information Fusion*, 8(4):387–412, 2007.

[42] A. Tahbaz-Salehi and A. Jadbabaie. A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4664–4669, 2006.

[43] N. Wilson. Algorithms for Dempster-Shafer theory. In D. M. Gabbay and P. Smets, editors, *Handbook of defeasible reasoning and uncertainty management. Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, pages 421–475. Kluwer Academic Publishers, Boston, 2000.

[44] F. Xiao. Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. *Information Fusion*, 46:23–32, 2019.

[45] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[46] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70, April 2005.

[47] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denœux. Multimodal information fusion for urban scene understanding. *Machine Vision and Applications*, 27:331–349, 2016.

[48] S. Zair and S. Le Hégarat-Mascle. Evidential framework for robust localization using raw GNSS data. *Engineering Applications of Artificial Intelligence*, 61:126–135, 2017.

[49] S. Zair, S. Le Hégarat-Mascle, and E. Seignez. An evidential RANSAC algorithm and its application to GNSS positioning. In J. Vejnarová and V. Kratochvíl, editors, *Belief Functions: Theory and Applications*, pages 241–250, Cham, 2016. Springer International Publishing.