

Low-Dimensional Topology of Information Fusion

Daniel Moskovich
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
dmoskovich@gmail.com

Avishy Y. Carmi
Faculty of Engineering Sciences
Ben-Gurion University of the Negev
Israel
avcarmi@bgu.ac.il

ABSTRACT

We provide an axiomatic characterization of information fusion, on the basis of which we define an *information fusion network*. Our construction is reminiscent of tangle diagrams in low dimensional topology. Information fusion networks come equipped with a natural notion of *equivalence*. Equivalent networks ‘contain the same information’, but differ locally. When fusing streams of information, an information fusion network may adaptively optimize itself inside its equivalence class. This provides a fault tolerance mechanism for such networks.

Categories and Subject Descriptors

H.1.1 [Systems and Information Theory]: General Systems Theory, Information Theory

Keywords

information fusion; information entropy; adaptive sensor networks; estimation theory; covariance intersection; fault tolerance; Reidemeister moves; tangle diagrams.

1. INTRODUCTION

Information fusion is an umbrella term for concepts and methodologies whose primary goal is to integrate heterogeneous pieces of information from diverse sources. An information fusion algorithm is a vital ingredient of virtually any autonomous system [5, 7, 9, 10]. Some other applications are sensor networks [11, 17], biometrics [15], and intelligent decision support systems [1, 18]. Here we consider pieces of information whose correlations are unknown and some of which may originate from unreliable sources. ‘Pieces of information’ may refer to estimates of unknown parameters or state variables, or to other related statistical measures such as (unnormalized) probability density functions, Fisher information and Shannon entropy.

We axiomatize information fusion (Definition 3.1), and we define an *information fusion network* (Definition 4.1). A similar concept of a *tangle machine* was defined in [2]. Information fusion networks admit a natural notion of equivalence (5). When one or more streams of information becomes faulty (*e.g.* biased or inconsistent), the faults will propagate differently in equivalent networks. At every moment, an information fusion network may be adaptively configured to its ‘least faulty’ equivalent configuration. Examples are given in Sections 2 and 6.

Our notion of equivalence parallels the notion of *ambient isotopy* in knot theory. As such, it represents a link between information fusion and low dimensional topology. As discussed in [3], low dimensional topology provides insight into the characteristic quantities or *invariants* of information fusion networks, such as how much information is required to uniquely specify the network (its *capacity*) and how many ‘independent sub-networks’ the network contains (its *complexity*).

It is a new idea to model data fusion networks using diagrammatic algebraic structures from low dimensional topology. Previous approaches have viewed such networks as directed graphs. Our approach emphasizes the distributive aspect of data fusion, which manifests itself as the ‘no double-counting’ axiom of Definition 3.1.

2. AN ILLUSTRATIVE EXAMPLE

2.1 The example

This example illustrates two different *information fusion networks* for fusing estimates. These networks are *equivalent* in the sense that any information in one network can uniquely be recovered from the information in the other, but they differ in the consistency of intermediate fused estimates. Thus, one network might be ‘better’ than the other.

Consider three estimators \hat{X}_0 , \hat{X}_1 and \hat{X}_2 for the same random variable X . We are also provided with estimators C_0 , C_1 , and C_2 for the error covariances:

$$\text{cov} [X - \hat{X}_i] \stackrel{\text{def}}{=} E [(X - \hat{X}_i)(X - \hat{X}_i)^T] - E [X - \hat{X}_i] E [X - \hat{X}_i]^T \quad i = 0, 1, 2, \quad (1)$$

where $E[\bullet]$ denotes the expectation taken with respect to the underlying joint probability distribution. The (j, k) th entry in $\text{cov} [X - \hat{X}_i]$ measures the correlation between the

errors in \hat{X}_i 's estimates of the j th and of the k th entries of the vector X .

A *consistent estimator* (\hat{X}_i, C_i) , also called a *conservative estimator*, is one which is not *too* optimistic about its belief of what the value of X is:

$$C_i \geq \text{cov} [X - \hat{X}_i] \quad (2)$$

i.e. the matrix difference $C_i - \text{cov} [X - \hat{X}_i]$ is positive semi-definite. Roughly speaking, this definition implies that a consistent estimator believes that the errors in its measurements of each entry in X are at least as strongly correlated as they actually are. We would like all estimators to be consistent because inconsistent estimators may diverge and may cause errors.

Our next goal is to fuse two consistent estimators (\hat{X}_i, C_i) and (\hat{X}_j, C_j) to obtain a new consistent estimator. The most common algorithm for fusing estimators is the Kalman filter, but to use it we need to know the cross-correlations between the estimators. These cross-correlations are typically unknown in practice. *Covariance intersection* (CI) provides a way around the problem, providing a method for fusing a pair of consistent estimates whose correlations are unspecified [6, 16]. The working principle of CI is that if \hat{X} and \hat{X}' are consistent estimators of X , then so is their convex combination (the proof is provided in [6]):

$$\hat{X}_a = (1-s)C_a C_0^{-1} \hat{X}_0 + s C_a C_1^{-1} \hat{X}_1, \quad (3a)$$

$$C_a^{-1} = (1-s)C_0^{-1} + s C_1^{-1}, \quad (3b)$$

where $s \in [0, 1)$ is a weight parameter. Different choices of the parameter s can be used to optimize the update with respect to different performance criteria.

Our next goal is to fuse **the triple** (\hat{X}_0, C_0) , (\hat{X}_1, C_1) , and (\hat{X}_2, C_2) to obtain a single consistent estimator for X . Two fusion schemes are:

- (A) First fuse (\hat{X}_0, C_0) with (\hat{X}_1, C_1) with a parameter s , then fuse the resulting estimator with (\hat{X}_2, C_2) with a parameter t .
- (B) First fuse both (\hat{X}_0, C_0) and (\hat{X}_1, C_1) with (\hat{X}_2, C_2) with the parameter t . Then fuse the resulting fused estimators with the parameter s .

Later, we will represent these two fusion schemes by Figure 1. Pairs are fused using CI using the same weights s and t . A short computation confirms that both of the above fusion schemes for a consistent triple (\hat{X}_0, C_0) , (\hat{X}_1, C_1) , and (\hat{X}_2, C_2) result in the same consistent estimator for X .

What if (\hat{X}_0, C_0) and (\hat{X}_2, C_2) are consistent, but (\hat{X}_1, C_1) is inconsistent? Then both fusion schemes **ultimately** lead to the same estimate of X , which is consistent for an appropriate choice of s and of t . But **at an intermediate stage**, the first fusion scheme involves the inconsistent estimate “ (\hat{X}_0, C_0) fused with (\hat{X}_1, C_1) with parameter s ”, whereas all intermediate fused estimates in the second scheme are

consistent. In this case the second fusion scheme is *better* than the first. Conversely, if (\hat{X}_1, C_1) is highly consistent compared to (\hat{X}_0, C_0) and (\hat{X}_2, C_2) , then the first fusion scheme would be better than the second.

2.2 The information fusion networks of our example

This section constitutes a preliminary discussion of the concept of an information fusion network in the context of our example, in anticipation of its definition in Section 4. An *information fusion network* is made up of labeled directed edges whose incident nodes may either be fixed points in the plane called *endpoints* or boxes called *interactions*. Each interaction involves one distinguished edge called the *agent* which goes ‘all the way though’ vertically and ‘emerges on the other side’ (*i.e.* we identify the edge incident to the top to the bottom of the interaction). Other edges are called *patients*, and come in pairs—there is one patient coming into the interaction for each patient going out of the interaction.

Figure 1 contains two subfigures, representing the two information fusion schemes in Section 2.1. Each have three *input edges*, labeled 0, 1, 2, denoting estimates (\hat{X}_0, C_0) , (\hat{X}_1, C_1) , and (\hat{X}_2, C_2) correspondingly, and a distinguished output edge labeled b in 1A and \bar{b} in 1B. Their interactions are labeled by operations \triangleright_s for some $s \in \mathbb{R}$, which represents an application of the fusion rule (3) with weight s . Explicitly, agents are weighted s and patients are weighted $1-s$.

A *fusion process* is a sequence of estimates stored in edges going from an initial to a terminal edge. Subfigure 1A contains three fusion processes, labeled $\{0, a, b\}$, $\{1, c\}$, and $\{2\}$. We write $0 \triangleright_s 1$ for the fusion of the estimate (\hat{X}_0, C_0) with (\hat{X}_1, C_1) according to the fusion rule (3) with a weighting parameter s . The resulting estimate (\hat{X}_a, C_a) is then passed on to the second interaction where it fuses with (\hat{X}_2, C_2) with parameter t , and resulting estimate $(0 \triangleright_s 1) \triangleright_t 2$ is stored in b .

The network of Subfigure 1B differs only in its first process, which reads:

$$0 \rightarrow \underbrace{0 \triangleright_t 2}_a \rightarrow \underbrace{(0 \triangleright_t 2) \triangleright_s (1 \triangleright_t 2)}_{\bar{b}} \quad (4)$$

Above we described a pair of equivalent information fusion networks which have different local performance. Alternating between these two networks to match a time evolution of initial estimator consistencies adaptively optimizes the consistency of the intermediate estimator at each point in time. In future sections we consider the general case.

3. ALGEBRA OF INFORMATION FUSION

Consider a sensor network [11, 17]. In distributed information fusion architectures, nodes behave as intelligent proxies fusing raw measurements streaming from their sensors with information received from neighboring nodes [17]. Fusion may be carried out within a node using a statistical filtering algorithm, *e.g.* the Kalman filter. Such algorithms normally use cross dependencies between the incoming pieces of information (*i.e.* raw measurements and estimates from

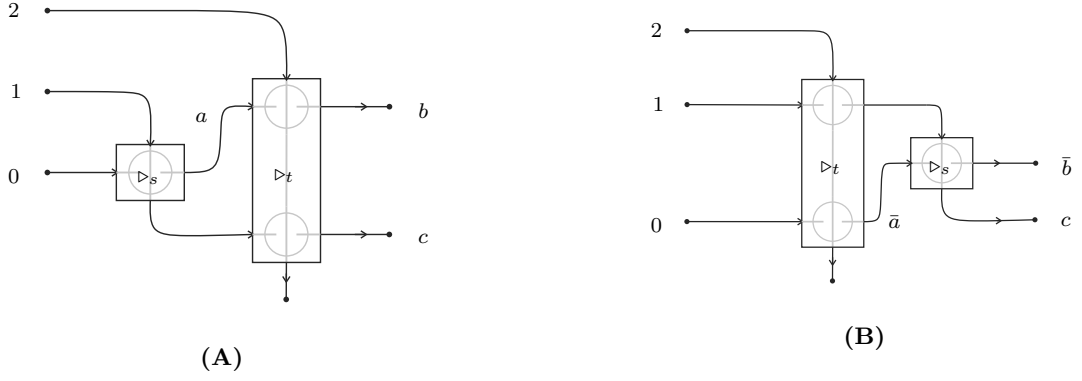


Figure 1. Subfigures 1A and 1B represent different schemes to fuse estimates (\hat{X}_0, C_0) , (\hat{X}_1, C_1) , and (\hat{X}_2, C_2) to a consistent estimate for X .

other nodes). However, large scale and complex networks generally inhibit calculations of the required statistical interdependencies among nodes [6]. Covariance intersection (CI) does not require knowledge of correlations, hence it is suitable for fusion in large scale settings [6, 16]. Definition 3.1 abstracts the key properties of CI to axiomatize information fusion. Chief among these is self-distributivity or *no double counting* which guarantees equality of outputs of the two networks of Section 2.

3.1 Algebraic axiomatization

In this section we formulate an algebra structure of ‘information’ subject to a binary ‘update’ operation. We name such a structure a *quandloid*, a portmanteau of “quandle” and “groupoid”.

DEFINITION 3.1 (QUANDLOID). A quandloid is a set Q , whose elements, called colours, represent pieces of information, together with a set B of partially-defined binary operations representing ‘updates’, satisfying the following properties:

- **Coherence of information.** A piece of information may update itself, and this neither generates new information nor loses information. Symbolically:

$$a \triangleright a = a \quad \forall a \in Q, \quad \forall \triangleright \in B. \quad (5)$$

- **Causal invertibility.** All update operations are left invertible. Thus, any input can uniquely be recovered from its corresponding output together with the agent. Symbolically, for every $\triangleright \in B$ there exists an ‘left-inverse operation’ $\triangleleft \in B$ such that:

$$(a \triangleright b) \triangleleft b = a \quad \forall a, b \in Q, \quad \forall \triangleright \in B. \quad (6)$$

More precisely, if $a \triangleright b$ exists then $(a \triangleright b) \triangleleft b$ exists and equals a .

- **No double counting.** Updating $a \triangleright_2 c$ by $b \triangleright_1 c$ with \triangleright_1 gives the same result as the updated piece of information $a \triangleright_1 b$ by c with \triangleright_2 . Thus c counts towards the final result only once, and there is no redundancy. Symbol-

ically:

$$(a \triangleright_1 b) \triangleright_2 c = (a \triangleright_2 c) \triangleright_1 (b \triangleright_2 c) \quad \forall a, b, c \in Q, \quad \forall \triangleright_1, \triangleright_2 \in B. \quad (7)$$

In particular, the left-hand side exists if and only if the right-hand side exists.

- **Identity.** The set B includes an ‘identity element’ \triangleright_e such that $a \triangleright_e b = a$ for all $a, b \in Q$.

By abuse of notation, we often write Q as a shorthand for (Q, B) .

Similar self-distributive structures have been studied in knot theory (e.g. [4]) and in the theory of computation (e.g. [14]). Specifically, if we were to require that all binary operations in B be fully defined—i.e. that $a \triangleright b$ exists for all $a, b \in Q$ and for all $\triangleright \in B$ then our notion would recover the notion of a multi-quandle [13].

3.2 Linear and log-linear quandloids

From Equation 3, we see that an example of a quandloid is the set Q of all pairs (\hat{X}, C) of an estimator \hat{X} for a random variable X with an estimator C for its error covariance matrix, whose update operations are \triangleright_s with $s \in [0, 1)$ and their formal inverses \triangleleft_s (where defined). Generalizing this example, we make a definition.

DEFINITION 3.2 (LINEAR QUANDLOID).

A linear quandloid is a real vector space \bar{Q} together with operations $\bar{\triangleright}_s$ such that:

$$a \bar{\triangleright}_s b \stackrel{\text{def}}{=} (1-s)a + sb, \quad a, b \in \bar{Q}, \quad (8)$$

with $s \in S \subset \mathbb{R}$ where $1 \notin S$ (by causal invertibility) but $0 \in S$ (by identity).

A second class of examples of quandloids is given below:

DEFINITION 3.3 (LOG-LINEAR QUANDLOID). Let Q be a space of measures defined over the σ -algebra of some set R ,

whose elements we think of as unnormalized probability density functions. Then equipping Q with the operations

$$p(x) \triangleright_s q(x) \stackrel{\text{def}}{=} p(x)^{1-s} q(x)^s, \quad p(x), q(x) \in Q, \quad (9)$$

again with $s \in S \subset \mathbb{R}$ where $1 \notin S$ (by causal invertibility) but $0 \in S$ (by identity) defines a quandle called a log-linear quandle.

Our next quandles will be derived from log-linear quandles using *homomorphisms*.

DEFINITION 3.4. A homomorphism of quandles is defined as a function $\mathbf{h} \stackrel{\text{def}}{=} (h_E, h_V): (Q, B) \rightarrow (\bar{Q}, \bar{B})$ such that

$$h_E(a \triangleright b) = h_E(a) h_V(\triangleright) h_E(b) \quad \forall a, b \in Q; \quad \forall \triangleright \in B. \quad (10)$$

3.3 Covariance intersection

The covariance intersection update rules are obtained via a homomorphism from a particular log-linear quandle. Explicitly:

PROPOSITION 3.5. Let

$$Q \stackrel{\text{def}}{=} \left\{ \exp \left(-\frac{1}{2} (X - \hat{X})^T C_X^{-1} (X - \hat{X}) \right) \mid \hat{X} \in \mathbb{R}^n, C_X \in \mathbb{R}^{n \times n} \right\} \quad (11)$$

be the space of unnormalized Gaussian probability density functions over \mathbb{R}^n . The underlying space of parameters $\bar{Q} = (\mathbb{R}^n, \mathbb{R}^{n \times n})$ together with the operations $\bar{\triangleright}_s: \bar{Q} \times \bar{Q} \rightarrow \bar{Q}$

$$\hat{Z} = (C_Z C_X^{-1} \hat{X}) \bar{\triangleright}_s (C_Z C_Y^{-1} \hat{Y}) = (1-s) C_Z C_X^{-1} \hat{X} + s C_Z C_Y^{-1} \hat{Y} \quad (12a)$$

$$C_Z^{-1} = C_X^{-1} \bar{\triangleright}_s C_Y^{-1} = (1-s) C_X^{-1} + s C_Y^{-1} \quad (12b)$$

for every $(\hat{X}, C_X), (\hat{Y}, C_Y) \in \bar{Q}$, with $s \in [0, 1]$, form the linear quandle of estimators and CI, considered at the beginning of this note.

The proof, which is straightforward, is deferred to the Appendix.

3.4 Fisher information

Let I_θ be the *observed* Fisher information matrix associated with the unnormalized joint probability density function:

$$(p \triangleright_s q)(X, \theta) \stackrel{\text{def}}{=} p(X, \theta) \triangleright_s q(X, \theta) \quad (13)$$

where θ is a random parameter vector whose prior is $g(\theta)$, and where $p(X, \theta) = p(X \mid \theta)g(\theta)$, $q(X, \theta) = q(X \mid \theta)g(\theta)$. Then I_θ satisfies:

$$\underbrace{\frac{\partial^2 \log(p \triangleright_s q)(X, \theta)}{\partial \theta_i \partial \theta_j}}_{I_\theta(p \triangleright_s q)} = \underbrace{\left(\frac{\partial^2 \log p(X, \theta)}{\partial \theta_i \partial \theta_j} \right)}_{I_\theta(p)} \bar{\triangleright}_s \underbrace{\left(\frac{\partial^2 \log q(X, \theta)}{\partial \theta_i \partial \theta_j} \right)}_{I_\theta(q)} \quad (14)$$

where:

$$a \bar{\triangleright}_s b \stackrel{\text{def}}{=} (1-s)a + sb, \quad a, b \in Q, \quad (15)$$

is a binary operation for a linear quandle. Note that, in the case of a random parameter vector, the normalizing factor is independent of the parameters, and hence we may use the unnormalized $p \triangleright_s q$ in the expression for the observed Fisher information matrix.

Note that *expected* Fisher information matrices do not form a quandle in general. Exceptional cases, when expected Fisher information matrices do form a quandle, are either when $I_\theta = E[I_\theta]$ or when the underlying expectation is taken exclusively with respect to the prior $g(\theta)$. We give examples of both of these cases.

Assume that $p(X, \theta) = p(X \mid \theta)g(\theta)$ and $q(X, \theta) = q(X \mid \theta)g(\theta)$ are colours of a log-linear Gaussian quandle Q , where θ is a mode whose prior $g(\theta)$ is also Gaussian. Then $h_E(\cdot) \stackrel{\text{def}}{=} I_\theta(\cdot)$ is a homomorphism and $(E[I_\theta], \bar{\triangleright}_s)$ is a linear quandle:

$$E[I_\theta(p \triangleright_s q)] = (C_p^{-1} + C_q^{-1}) \bar{\triangleright}_s (C_p^{-1} + C_q^{-1}) = E[I_\theta(p)] \bar{\triangleright}_s E[I_\theta(q)] \quad (16)$$

where C_p, C_q and C_g denote the covariances of p, q and g , and $s \in [0, 1]$.

Here is another example. Consider a log-linear quandle whose colours are of the form $p(X, \theta) = p(X \mid \lambda_p(\theta))g(\theta)$ and $q(X, \theta) = q(X \mid \lambda_q(\theta))g(\theta)$ where the underlying conditionals are exponential densities whose θ -dependent rate parameters are λ_p and λ_q . As before $g(\theta)$ denotes the prior of θ . Additionally, we assume that the second derivative of any rate parameter vanishes $d^2 \lambda(\theta)/d\theta^2 = 0$. In such a case, indeed:

$$E[I_\theta(p \triangleright_s q)] = E[I_\theta(p)] \bar{\triangleright}_s E[I_\theta(q)]. \quad (17)$$

and thus $(E[I_\theta], \{\bar{\triangleright}_s\})$ a linear quandle.

3.5 Shannon information

This example was discussed in [2].

A binary information source is described by a Bernoulli random variable X . The entropy of X , denoted $H(X)$, has an operational meaning given by Shannon's source coding theorem. Very long independent identically distributed sequences generated by such a source fall into two categories: they are either *typical* or not. The probability of a typical sequence stabilizes around the value $2^{-NH(X)}$ which implies that not more than $NH(X)$ bits are required to encode any typical message with 'negligible' loss of information.

Let Q be a log-linear quandle whose colours are uniform probability densities of typical sequences associated with infinitely many information sources

$$Q \stackrel{\text{def}}{=} \left\{ 2^{-NH(X)} \mid X \text{ is an information source} \right\} \quad (18)$$

The set of entropies $\{H(X)\}$ with operations $\bar{\triangleright}_s$ constitute a linear quandle:

$$H(Z) = H(X) \bar{\triangleright}_s H(Y) = (1-s)H(X) + sH(Y), \quad H(X), H(Y) \in \mathcal{H} \quad (19)$$

which follows from the homomorphism $h_E(\cdot) \stackrel{\text{def}}{=} -\frac{1}{N} \log(\cdot)$.

An update in this quandle may describe a source Z which behaves like X in $(1-s)$ fraction of times and like Y in the rest. For example, generating a message of N symbols from Z , one expects that approximately $(1-s)N$ of them will make up a typical sequence from X whereas the remaining sN symbols will look as though they were generated by Y . In other words, the source Z concatenates parts of the message from X with parts of the message from Y .

4. WHAT IS AN INFORMATION FUSION NETWORK?

We assemble information updates into a network.

DEFINITION 4.1 (INFORMATION FUSION NETWORK).

An information fusion network is:

- A directed graph G whose vertices (drawn as boxes e.g. in Figure 1) have either degree 1 or have positive even degree. Vertices of even degree are called interactions, and the in-degree of each interaction equals its out-degree. Vertices of degree 1 are called endpoints. An endpoint is initial if it is a source, and is terminal if it is a sink.
- For each interaction v , whose degree we denote $2m+2$, a partition of edges incident to v into pairs:

$$\{(f^{in}, f^{out})\} \cup \{(e_1^{in}, e_1^{out}), (e_2^{in}, e_2^{out}), \dots, (e_m^{in}, e_m^{out})\}. \quad (20)$$

We consider each of the pairs (f^{in}, f^{out}) as though it were a single edge, and refer to that pair as the agent of v , and we call each e_i^{in} an input of v , and its corresponding edge e_i^{out} an output of v . All edges with superscript ‘in’ (i.e. f^{in} and $e_1^{in}, e_1^{in}, \dots, e_m^{in}$) are directed towards v , and all edges with superscript ‘out’ are directed away from v .

- A colouring function

$$\rho \stackrel{\text{def}}{=} (\rho_E, \rho_V): E(G) \times V(G)_{\text{deg} \geq 2} \rightarrow Q \times B. \quad (21)$$

Function ρ must satisfy:

- If (f^{in}, f^{out}) is the agent of v , then $\rho_E(f^{in}) = \rho_E(f^{out})$.
- For each input-output pair (e_i^{in}, e_i^{out}) we have either $\rho_E(e_i^{out}) = \rho_E(e_i^{in}) \triangleright \rho_E(f)$ or else $\rho_E(e_i^{in}) = \rho_E(e_i^{out}) \triangleright \rho_E(f)$ where \triangleright denotes $\rho_V(v)$, and where $\rho_E(f) = \rho_E(f^{in}) = \rho_E(f^{out})$ is the colour of the agent of v .

Each transition $\rho_E(e_i^{in}) \mapsto \rho_E(e_i^{out})$ is called an update if $\rho_E(e_i^{out}) = \rho_E(e_i^{in}) \triangleright \rho_E(f)$ and is called a discount if $\rho_E(e_i^{in}) = \rho_E(e_i^{out}) \triangleright \rho_E(f)$.

Our diagrammatic convention is to update from right to left of the agent edge(s), so that if the agent edge is drawn pointing from top to bottom then the colour of an edge to the left of a box is updated by \triangleright to become the colour of the corresponding edge the right of the box, and if the agent edge is drawn pointing from bottom to top then the opposite.

Note that the number of initial endpoints of an information fusion network equals its number of terminal endpoints. This relates information fusion to reversible computation, thanks to causal invertibility [8].

If (Q, B) were a multi-quandle, then the definition of an information fusion network would be equivalent to the definition of a tangle machine in [2].

5. NETWORK EQUIVALENCE

In this section we describe three local modifications on information fusion networks. These *Reidemeister moves* are drawn in Figures 2 and 3. Two information fusion networks which differ by a finite sequence of these modifications are considered to be *equivalent*. For this reason, we think of these modifications as *conservation laws*.

- *Reidemeister I*. This local move modifies a network by eliminating or introducing an update of an edge by itself. By coherence of information, no information is gained or lost when we perform this move.
- *Reidemeister II*. This local move modifies a network by updating an edge, and immediately discounting it by the same agent. By causal invertibility doing such a thing does not effect the colour of the edge, and so no information is gained or lost when we perform this move.
- *Reidemeister III*. This local move modifies a network by replacing updated outputs with a common agent by corresponding updated inputs updated by the correspondingly updated agent. By ‘no double counting’ doing such a thing does not effect the colour of the edge, and so no information is gained or lost when we perform this move.

In low dimensional topology, the *Reidemeister Theorem* tells us that two diagrams of tangles are related by a finite sequence of Reidemeister moves if and only if the tangles which they represent are *ambient isotopic*. Combined with a result stating that a tangle is equivalent to the set of all of its colourings, tells us that the Reidemeister moves are the unique set of local moves on tangles which preserve the information content of the tangle—i.e. the set of all possible tangle colourings. The appearance of Reidemeister moves signifies that the theory of information fusion networks has a low dimensional topological flavour.

We believe that the Reidemeister moves are diagrammatic representations of fundamental symmetries of information fusion (although there may also be other symmetries), and therefore that the formalism of information fusion networks subject to Reidemeister moves is a topological candidate for a suitably flexible diagrammatic language with which to discuss adaptive networks of information fusion.

6. TOPOLOGICAL RESILIENCE

In many applications, raw data is acquired from external sources (e.g. from sensors) over a period of time. In such cases, statistical properties of the input information are time-dependent. Suppose that we are given many such information input streams, and that our goal is to fuse some of

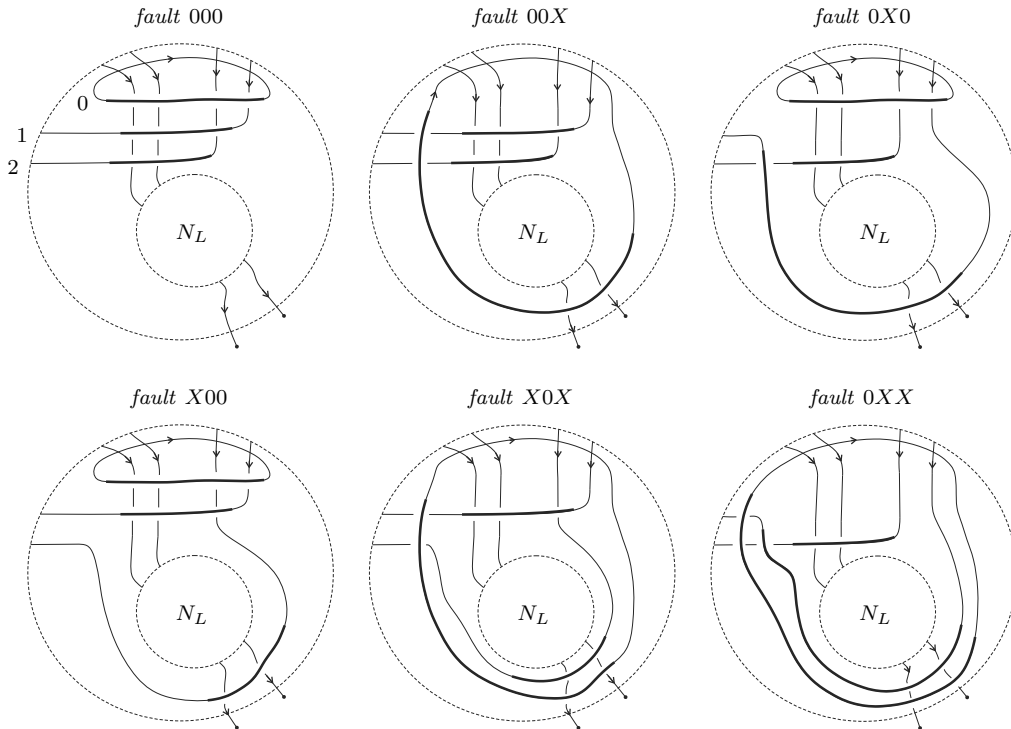


Figure 4. Topological fault-tolerance. Equivalent networks ameliorate the effects of various occurring faults. The fault code appearing above each network should be read as follows. The digits from left to right correspond to edges 2,1 and 0. An “X” in the place of the i th digit signifies faulty content in its respective edge.

them, maybe to discount others (perhaps in order to isolate the contribution of one set of streams over another, or perhaps for a different reason), and ultimately to generate some set of output streams. In doing so, we form a complicated information fusion network whose interaction parameters we compute to optimize some global objective function.

For example, in a network coloured by a log-linear quandle, we may choose the *Chernoff information* as our global objective [12]. To find optimal parameters $\{s_i\}$, compute:

$$\min_{\{s_i\} \in (0,1)} \left[\sum_{\{j \text{ is an output}\}} \log \int_x p_j(x) dx \right], \quad (22)$$

where $p_j(x)$ denotes the unnormalized probability density in the j -th output.

The global goal of the network is to generate its output colours. However, the results of intermediate computations may also matter to us, for example in order to backtrack to correct an error. Reidemeister moves are local, so they change only a small part of the network, with the rest of the network remaining unchanged. At every moment the network has a ‘best’ representation, which is the representative in our equivalence class of information fusion networks for which some chosen local objective function is maximized. The network may adaptively be configured as follows:

- i. Optimize the network quandle parameters $\{s_j\}$ with respect to some global objective function (e.g. (22)). This defines an equivalence class of networks $[T]$.

- ii. Adaptively optimize the network $T' \in [T]$ within its equivalence class with respect to some local objective function.

The algorithmic aspect of designing such a topological fault-tolerance scheme to optimize a network with respect to a given local objective function might be complicated in general. To illustrate the utility of our theory, we give an example for which it is easy. Consider the network illustrated in the upper left corner of Figure 4. This network has a set of intermediate edges lies inside the subnetwork designated by N_L and represented as an empty circle.

In the course of network operation, erroneous streams of data cause one or more of the edges 0,1, and 2, to carry faulty pieces of information, e.g. biased, inconsistent and otherwise unreliable estimates. This is detected inside the network. To inhibit the influence of this contamination on edges within N_L , the network performs Reidemeister moves on itself, transforming itself into an equivalent network. This is achieved by ‘sliding’ the faulty edges all the way over N_L , by repeated application of the second and third Reidemeister moves. In the resulting topology, the faulty edges have no effect on N_L , and so local costs are improved.

7. CONCLUSION

In this note, we provided an axiomatic characterization of information fusion as an operation of a *quandle*. The key properties of information fusion are no double counting (distributivity) and causal invertibility. We showed that covari-

ance intersection of estimators, fusion of observed Fisher information matrices, and fusion of Shannon information are all quandle operations, and that they are all obtained from a single log-linear quandle by means of homomorphisms. Inspired by the low-dimensional topological theory of tangle diagrams, we defined an *information fusion network*, which comes equipped with a natural notion of equivalence. Two information fusion networks are equivalent if they differ by a finite sequence of Reidemeister moves. Our examples demonstrated that information fusion networks are fault-tolerant, in the sense that they can adaptively be optimized to minimize the effect of faulty input information on the network.

APPENDIX

Proof of Proposition 3.5

The covariance intersection update rules are obtained via a homomorphism from a particular log-linear quandle.

PROOF. Let $p(X), p(Y) \in Q$ be unnormalized Gaussian densities parameterized by their means and their covariances, (\hat{X}, C_X) and (\hat{Y}, C_Y) . An update operation reads:

$$\begin{aligned} p(X) \triangleright_s p(Y) &= p(Z) \\ &= \exp \left(-\frac{1}{2} \left[(1-s)(Z - \hat{X})^T C_X^{-1} (Z - \hat{X}) \right. \right. \\ &\quad \left. \left. + s(Z - \hat{Y})^T C_Y^{-1} (Z - \hat{Y}) \right] \right) \end{aligned} \quad (23)$$

We verify that this expression equals:

$$\exp \left(-\frac{1}{2} (Z - \hat{Z})^T C_Z^{-1} (Z - \hat{Z}) \right), \quad (24)$$

for \hat{Z} the form which appears in (12a) and for C_Z of the form which appears in (12b).

Given $p(X) \in Q$, we compute that $C_X^{-1} = -\frac{\partial^2 \log p(x)}{\partial x \partial x^T}$, which is called the *precision matrix* of $p(X)$. In particular,

$$C_Z^{-1} = -\frac{\partial^2 \log p(z)}{\partial z \partial z^T} = (1-s)C_X^{-1} + sC_Y^{-1} = C_X^{-1} \bar{\triangleright}_s C_Y^{-1} \quad (25)$$

which follows from (23).

For any $p(X) \in Q$ the mode \hat{X} satisfies:

$$\left. \frac{\partial \log p(x)}{\partial x} \right|_{x=\hat{X}} = 0 \quad (26)$$

The mode \hat{Z} of $p(Z)$ is thus obtained from (23) by

$$2 \left. \frac{\partial \log p(z)}{\partial z} \right|_{z=\hat{Z}} = (1-s)C_X^{-1}(\hat{Z} - \hat{X}) + sC_Y^{-1}(\hat{Z} - \hat{Y}) = 0 \quad (27)$$

finally yielding

$$C_Z^{-1} \hat{Z} = (1-s)C_X^{-1} \hat{X} + sC_Y^{-1} \hat{Y} = \left(C_X^{-1} \hat{X} \right) \bar{\triangleright}_s \left(C_Y^{-1} \hat{Y} \right). \quad (28)$$

□

A. REFERENCES

- [1] Bass, T. 2000 Intrusion detection systems and multisensor data fusion. *Communications of the ACM* **43**(4), 99–105.
- [2] Carmi, A.Y. Moskovich, D. 2014 Tangle machines I: Concept. Manuscript submitted for publication. <http://arxiv.org/abs/1404.2862>
- [3] Carmi, A.Y. Moskovich, D. 2014 Tangle machines II: Invariants. Manuscript submitted for publication. <http://arxiv.org/abs/1404.2863>
- [4] Carter, S.J. 2009 A survey of quandle ideas. In *Introductory lectures on Knot Theory: Selected lectures presented at the Advanced School and Conference on Knot Theory and its Applications to Physics and Biology*, 22–53. <http://arxiv.org/abs/1002.4429>
- [5] Hall, D.L. Llinas, J. 1997 An introduction to multisensor data fusion. *Proceedings of the IEEE* **85**(1), 6–23.
- [6] Julier, S. Uhlmann, J.K. 2001 General decentralized data fusion with covariance intersection. In *Handbook of Multisensor Data Fusion*, CRC Press, Boca Raton, FL.
- [7] Khaleghi, B. Khamis, A. Karray, F.O. Razavi, S.N. 2013 Multisensor data fusion: A review of the state-of-the-art. *Information Fusion* **14**(1), 28–44.
- [8] Landauer, R. 1961, Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, vol. 5, 183–191.
- [9] Lou, R.C. Chih-Chen, Y. Kuo, L.S. 2002 Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal* **2**(2), 107–119.
- [10] Murphy, R. 2000 Introduction to AI robotics. *MIT Press*
- [11] Nakamura, E.F. Loureiro, A.F. Frery, A.C. 2007 Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys* **39**(3)
- [12] Nielsen, F. 2013 An information–geometric characterization of Chernoff information. *IEEE Signal Processing Letters* **20**(3), 269–272.
- [13] Przytycki, J.H. 2011 Distributivity versus associativity in the homology theory of algebraic structures. *Demonstration Mathematica* **44**(4), 823–869. [arXiv:1109.4850](https://arxiv.org/abs/1109.4850)
- [14] Roscoe, A.W. 1990 Consistency in distributed databases. *Oxford University Computing Laboratory Technical Monograph PRG-87*.
- [15] Ross, A. Jain, A. 2003 Information fusion in biometrics. *Pattern Recognition Letters* **24**(13), 2115–2125.
- [16] Uhlmann, J. K. 2003 Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion* **4**, 201–215.
- [17] Xiong, N. Svensson, P. 2002 Multi-sensor management for information fusion: issues and approaches. *Information Fusion* **3**, 163–186.
- [18] Yong, D. WenKang, S. ZhenFu, Z. Qi, L. 2004 Combining belief functions based on distance of evidence. *Decision Support Systems* **38**(3), 489–493.