Efficient and Accurate Encodings for Subjective Logic Opinions

Master Thesis Proposal

2024-08-08

Sophie Hirn
sophie.hirn@uni-ulm.de

Ulm University

Source: https://cgit.sowophie.io/master/
Latest version: https://artifacts.sowophie.io/master/proposal.pdf

This work is licensed under Creative Commons Attribution 4.0 International. To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/.

1. Motivation

1.1. Research Field

The vision of the "smart" car, that drives itself and is connected to its environment, has inspired a lot of research activity in the past decade. In scientific literature, this is often referred to as **Connected, Cooperative and Autonomous Mobility (CCAM)**. This broad term entails many different technologies for different scenarios, such as:

- **Cooperative Intersection Management (CIM)**[1], the managing of an intersection using data sent from nearby vehicles to improve throughput and safety
- **Cooperative Adaptive Cruise Control (CACC)**, vehicles forming so-called platoons to efficiently manage their road speed together

The underlying networking technology for this is called **Vehicular Ad-Hoc Networks (VANETs)**, i.e. networks that form and dissolve organically around vehicles. They naturally deal with data from mostly untrusted sources, on which nodes must base their decisions, which in turn can affect the physical wellbeing of humans in and around the vehicles significantly. While it generally can be assumed that most nodes inside a VANET are not malicious[2], the consequences of system failures include threat to human lives, and therefore an abundance of caution is indicated. Dangerous data does not have to originate from a malicious attacker, but could even be emitted by a faulty component.

Technologies to tackle this problem are grouped under the umbrella term **Misbehavior Detection (MBD)**. Many systems are based on establishing and maintaining representations of *trust* in a node and/or its sensors. While susceptible to sudden, unannounced attacks, they can reliably deal with clients occasionally or consistently sending wrong data[3].

1.2. Thesis Area

Subjective logic[4] is a framework for modelling subjective, distributed uncertainty, i.e. situations where nodes have different views of probabilities on other nodes. It has recently gained popularity for modelling trust in CCAM scenarios.

Trust as described by subjective logic contains of a 4-tuple of probabilities (b, d, u, α). b and d represent the expected probability of some statement being correct and incorrect respectively. Together with u, the uncertainty, they add up to 1. α represents the expected base probability of the statement being true in the absence of any evidence. Given this tuple, the combined probability of the statement X being true is:

$$P(X) = b + u\alpha$$

Trust opinions can also be combined using a number of operators. Different operators are useful in different scenarios, and there is no single "correct" operator that models every application well[5].

Subjective logic as introduced here is only able to represent *binomial* probability distributions, i.e. those with two discrete outcomes, usually *true* and *false*. While extensions of this model exist to represent general probability distributions, e.g. imprecise sensors with normal error distribution, most literature on CCAM uses the simple binomial model, so we shall do the same.

One big consideration with CCAM models is the representation of trust opinions. Any form of communication has bandwidth limits. With wireless communication, it is especially important to be mindful of them as messages are usually broadcasted and the limit is global for a given network. In turn, this means that the more active nodes join network, the less bandwidth is available for every individual node. The resulting desire to keep CCAM messages as short as possible is opposed by the desire to have calculations be as accurate as possible. And the simplest way to do this is to increase the number of bits dedicated to the representation of data. This thesis will explore this dilemma, and try to find ways to provide satisfactory calculation precision within capacity limits.

2. State of the Art

2.1. Operators for Combining Trust Opinions

Jøsang, who first described subjective logic in his book[4], describes a number of different operators for combining trust opinions[5]. Based on these operators, Cheng et al. describe a system of operations[3] that can be used to:

- Combine a short-term observation into a long-term trust opinion (*Cumulative Fusion*)
- Combine a chain of two opinions into a transitive opinion (*Discounting Operator*)
- Combine two opinions from different perspectives into a single opinion (*Averaging Fusion*)

These operators are of special interest to us when it comes to evaluating the accumulation of errors in the system.

2.2. Number Representations

Many computer based subjective logic models use **Floating Point (FP)** numbers to store and calculate their probabilities. This is the preferred general-purpose format for storing non-integral numbers on computers. It has been standardized in IEEE 754[6], and implemented in the hardware of most modern CPUs.

FP numbers are stored as a packed triplet of values, (s, e, m). s represents the *sign* of the result, and is either 0 or 1. e is the *exponent* of a pre-selected base b ($= 2$ in most cases), and m is the so-called *mantissa* with $1 \leq m < b$. The encoded value f is defined as

$$f = (-1)^s \cdot m \cdot b^e$$

Apart from this definition for "normal" numbers, there are special values for very small values ("subnormals"), "not a number", and positive and negative infinity.

IEEE 754 defines a number of FP formats with various value ranges for m and e, allowing for a tradeoff between representable range and accuracy, and space usage. While FP numbers can represent a far wider range of values than two's complement integers with the same storage size, they are not magic, and the tradeoff for this flexibility is a loss of precision. They can represent some numbers exactly, but are often just approximations. One can find many

examples of them violating mathematical identities without too much effort.

The distance between two adjacent replaceable numbers is called the **quantum** ε, and for FP numbers it is dependent on e (unlike with traditional integers, where the quantum is a constant 1).

While FP numbers are a very fast format for performing calculations on most modern processors, they are by no means optimal for representing trust opinions. As the numbers in our models are always between 0 and 1, the sign is always positive. Even worse, the large range of the exponent is mostly going to waste. There is no need to represent any number greater than 1, and the smaller numbers are only of limited use to us as well.

A different solution for encoding non-integral numbers are so-called **fixed point** numbers. They consist of a single value n which is interpreted as

$$f = n \cdot 2^{-e}$$

for a pre-determined exponent $e \geq 0$, with the trivial case $e = 0$ representing traditional integers. Like integers and unlike floating point numbers, fixed point numbers have a constant quantum over their entire value range.

2.3. Interval Arithmetic

A natural approach for quantifying the error of calculations is **interval arithmetic**, in which calculations are performed on intervals of values at once, yielding result intervals. We can make initial estimations for the error interval of a number representation using its quantum. The accumulation of errors across complicated calculations can then be traced by using interval arithmetic.

For FP numbers, the error introduced by each primitive operation has been characterized by Hickey et al.[7]. The authors show that a closed algebra can be formed by using a pair of FP lower and upper bound values, and that the basic mathematical operators can be efficiently implemented using just basic IEEE 754 operations and a few case distinctions. They further prove that the result is correct and optimal with regard to FP limitations, i.e. the FP results are always a superset of the actual theoretical results, and the FP bounds are as close to the actual bounds as the FP representation allows them to be.

Fixed point numbers can be analyzed with this method as well, by treating them as a special case of floating point numbers with fixed exponent.

3. Problem Statement

3.1. Thesis Focus

The goal of this thesis is to find efficient representations to transmit trust opinions through channels with limited capacity. The trade-off between loss of precision and used bandwidth is to be characterized and discussed, if possible using the simulation results from ... For the representations considered, the error accumulation is to be characterized using interval arithmetic.

Disregarding the constant α for a moment, we can represent the tuple (b, d, u) with just the pair (b, d) using the model invariant $b + d + u = 1$. This allows us to make the message significantly smaller. Furthermore, we avoid violations of this invariant that arise from tracking all three values explicitly in a system with floating point errors.

Fixed Point Numbers

As outlined above, we can also forego encoding the exponent and sign bits of the floating point representation, leaving us with a pair of fixed point numbers. The width of the number determines the precision of the representation, and can be varied according to precision and bandwidth requirements. Characterizing the impact of this parameter is one of the goals of the thesis.

This leaves us with the choice of the exponent, and there are two natural choices, each with their own drawbacks. Let us assume the fixed point number

$$f = n \cdot 2^{-e}$$

is represented using k bits, i.e.

$$0 \leq n \leq 2^k - 1$$

If we pick $e = k$, we get values from 0 to

$$\begin{aligned} f_{\max} &= n_{\max} \cdot 2^{-e} \\ &= (2^k - 1) \cdot 2^{-k} \\ &= 2^k \cdot 2^{-k} - 1 \cdot 2^{-k} \\ &= 2^{k-k} - 2^{-k} \\ &= 1 - 2^{-k} \end{aligned}$$

This representation gives us a very good mapping of the range between 0 and 1, but cannot represent an exact 1. A model based on this would have to live with never being able to represent complete belief or disbelief.

Alternatively, we could shift the virtual point one bit to the right, with $e = k - 1$. This would yield a range from 0 to

$$\begin{aligned} f_{\max} &= n_{\max} \cdot 2^{-e} \\ &= (2^k - 1) \cdot 2^{-(k-1)} \\ &= 2^k \cdot 2^{-(k-1)} - 1 \cdot 2^{-(k-1)} \\ &= 2^{k-(k-1)} - 2^{-(k-1)} \\ &= 2 - 2^{-(k-1)} \end{aligned}$$

With this representation, we could comfortably represent the range from 0 to 1 inclusive, but have halved precision compared to the last model. This trade-off is to be characterized in the thesis.

It is important to stress that the formats described in this thesis are intended merely as efficient transfer encodings. The computers processing them do not have to use this as their internal representations. In fact, they can use their built-in floating point formats for efficient calculations. Conversions between floating and fixed point numbers can be implemented efficiently using simple bit shifts. They even incur no loss of precision beyond the differences of representable numbers defined by their respective bit widths.

Differential Updates

If the message header still has at least one unused bit left to encode different kinds of messages, we can save some additional bandwidth using an expected characteristic: subsequent trust opinions on the same subject are in most cases bound to be similar, i.e. have a small difference. We could define an "update" packet that transmits a (signed) difference between the old and the new value. By choosing an aggressive e, we can define a format that can efficiently represent small differences in much less bits than a full message, without sacrificing precision. Conceptually, we are just not transmitting the leading bits of the difference, which are going to be zero.

If the sender finds that the difference between old and new value is too big to fit into a update message, it can just transmit regular message with the full format instead.

This could even be used to transmit a value of 1 in a system that does not naturally support that in its encoding, as outlined above. A 1 would be sent by first transmitting a trust opinion of the highest representable value, then an update adding the missing part to the value.

Unfortunately, this only works reliably in systems which guarantee that each message is received exactly once by each intended recipient. Wireless networks are usually not communication channels which have this property. But if a transmission control scheme, such as TCP, is already used on top of the wireless network for other reasons, we can take advantage of that. If not, adding the scheme purely to facilitate differential updates is probably not going to have a positive impact on the amount of traffic sent.

3.2. Research Questions

The thesis will aim to answer the following questions:

1. How can a trust opinion be packed efficiently into a given packet size?
2. How does the transfer representation affect calculcation precision?
3. How can these calculations be made safely, i.e. without the errors leading to an overestimation of model security?

4. Approach

It is not hard to show that a fixed point representation with $e = k$ is the optimal representation for numbers $f \in [0, 1)$ with uniform coverage.

The basis for our precision estimations are going to be the combining operators defined by Cheng et al.[3]. Using an interval arithmetic representation, we can map uncertainty before the operation to uncertainty after the operation. Given this formula, we can then estimate the loss of precision in different scenarios, and quantify the tradeoffs introduced in section 3. Of special interest are encodings that fit common word sizes, e.g. 8, 16, 32, 64, ... bits per message.

We can also try to derive modified operators that provide conservative estimates, with the inherent loss of precision folded into the model uncertainty. Finding a way to do this without having to track the individual error intervals of each value would be hugely beneficial.

Finally, given the expected distribution of values in the field, we can see if statistical encoding schemes can be used to reduce the message size even further, without sacrificing numeric precision.

5. Planning

5.1. Own Background

The author has extensive experience with compilers, processors, and low-level data representations. This proficiency is the result of both university lectures, projects, and private experiments. The master specialization "IT Security" is not directly relevant for the core of this thesis, but helps to establish the subject into a broader context.

5.2. Work Packages

For each package, the following columns are given:

- Duration in weeks
- Earliest possible start in week no.
- Earliest possible end before week no.
- Latest possible start in week no.
- Latest possible end in week no.

The entire plan is calculated with four weeks of slack, i.e. if delays of no less than four weeks arise, the plan can still be executed without reconsideration. Work packages marked with † are not mission critical and could be shortened or dropped if the thesis falls behind schedule.

Work package	Dur	ES	EE	LS	LE
Work out term for error range of trust operators	6	1	7	5	11
Calculate error values for different message sizes	1	7	8	11	12
Derive "safe" belief fusion operators	2	8	10	12	14
Numeric long term simulation †	4	10	14	14	18
Evaluation	2	14	16	18	20
Finalize thesis	5	16	21	20	25

5.3. Contingency Plan

Some work packages, especially the first one, are hard to quantify upfront. The plan has been calculated with two weeks of slack, and the packages have also been given a safety margin. However, this may not be enough, and the project might fall behind schedule more than planned for.

In this case, some less critical work packages, as defined above, can be omitted without threatening the core subject of the thesis. This can help get the thesis back on track.

References

[1] Frank Kargl, Nataša Trkulja, Artur Hermann, Florian Sommer, Anderson Ramon Ferraz de Lucena, Alexander Kiening, and Sergej Japs, "Securing Cooperative Intersection Management through Subjective Trust Networks," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring),* IEEE (2023).

[2] Randall Munroe, "Self-Driving Issues," *XKCD*(1958).

[3] Mingxi Cheng, Chenzhong Yin, Junyao Zhang, Shahin Nazarian, Jyotirmoy Deshmukh, and Paul Bogdan, "A General Trust Framework for Multi-Agent Systems," *AAMAS '21: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*(20), ACM (2021).

[4] Audun Jøsang, *Subjective Logic: A Formalism for Reasoning under Uncertainty,* Springer International Publishing AG (2016). ISBN: 978-3-319-42335-7.

[5] Audun Jøsang, "Categories of Belief Fusion," *Journal of Advances in Information Fusing* **13**(2) (2018).

[6] "IEEE Standard for Floating-Point Arithmetic" in *IEEE Std 754-2019* (2019).

[7] T. Hickey, Q. Ju, and M. H. Van Emden, "Interval arithmetic: From principles to implementation," *Journal of the ACM* **48**(5), ACM (2001).