How to Represent Uncertainty via Qudits: Probability Distributions, Regular, Intuitionistic, Pythagorean, and Picture Fuzzy Sets, F-Transforms, etc.

Olga Kosheleva and Vladik Kreinovich
University of Texas at El Paso
El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu

Abstract

While modern computers are fast, there are still many important practical situations in which we need even faster computations. It turns out that, due to the fact that the speed of all communications is limited by the speed of light, the only way to make computers drastically faster is to drastically decrease the size of computer's components. When we decrease their size to sizes comparable with micro-sizes of individual molecules, it becomes necessary to take into account specific physics of the micro-world – known as quantum physics. Traditional approach to designing quantum computers – i.e., computers that take effect of quantum physics into account – was based on using quantum analogies of bits (2-state systems). However, it has recently been shown that the use of multi-state quantum systems – called qudits – can make quantum computers even more efficient.

When processing data, it is important to take into account that in practice, data always comes with uncertainty. In this paper, we analyze how to represent different types of uncertainty by qudits.

1 Outline

Why qudits: in brief. The need for faster computations necessitates the need to make computer components smaller and smallest – and the smaller we make them, the more important is to take quantum effects into account. From this viewpoint, quantum computing – computing on devices for which we need to take quantum effects into account – is inevitable. Traditional quantum computing techniques are based on qubits – quantum analogues of 2-state components (bits). However, lately, it has been shown that it is often beneficial to use quantum analogues of d-state components for $d > 2$. Such analogues are known as

qudits.

Why uncertainty: in brief. Input to computations comes from measurements and expert estimates. In both cases, the values we submit to algorithms are known with uncertainty. In this paper, we analyze how different types of uncertainty can be represented in the qudit form.

The structure of the paper. The structure of the paper is as follows. In Section 2, we explain how the need to speed up computations leads to quantum computing. In Section 3, we explain what are qudits, and in Section 4, we explain how qudits can be used to represent different types of uncertainty.

2 Why Quantum Computing

We need faster computers. While modern computers are very fast, there are still many practical problem for which their computation speed is not enough. An example of such a problem is tornado prediction. Just like we can reasonably accurately predict tomorrow's weather – by spending several hours on a high-performance computer, we can also predict, by spending the same computation time, in what direction a tornado will turn in the next 15 minutes. For predicting weather, several hours of computing still result in a prediction being ahead of the actual event, but for tornados, this computation time makes no sense: by the time we have our predictions, the tornado will already have turned.

To solve such problems, we need faster computations.

To make computers faster, we need to make their components smaller. How can we make computers faster? The speed of current computers is limited – somewhat surprisingly – by fundamental physics: namely, by the fact that, according to physics, no process can be faster than the speed of light; see, e.g., [7, 22]. For a usual laptop of approximately 30 cm size, the fastest way to send a signal from one of its sides to another one can be obtained if we divide 30 cm by the speed of light – which is approximately 300 000 km/sec. As a result, we get one nanosecond – 10^{-9} of a second. During this time, a usual 4GHz computer – i.e., a computer that performs 4 operations per nanosecond – will already perform 4 operations.

From this viewpoint, the only way to drastically speed up computations is to drastically shrink the computer – and thus, to drastically shrink all its components.

Enter quantum effects. The current computer cells are already almost the size of a few thousands of molecules. So if we drastically shrink them, their size will be comparable to a molecule size. To describe objects at such micro-size, it is no longer sufficient to use the usual Newton's mechanics – it is necessary to take into account effects of quantum physics – the physics of micro-world; see, e.g., [7, 22]. And this is exactly what is called *quantum computing* – computing with devices whose performance cannot be described without taking quantum effects into account.

Quantum computing: from necessary evil to spectacular (and scary) promises. At first, computer engineers viewed these quantum effects purely negatively. For example, in quantum physics, results can only predicted with some probabilities, which is a big problem when we want to design a computer that returns the same (correct) answer every time. However, later, scientists learned how to "tame" these probabilities and come up with deterministic devices.

Moreover, it turns out that in many cases, the specific formulas describing quantum-based probabilistic uncertainty can speed up computations even more; see, e.g., [12]. For example, by using quantum computing, we can find, in an unsorted n-element list, an element with a desired property in time proportional to $\sqrt{n}$ – which in non-quantum case, we cannot do it faster than in n computational steps: indeed, if we do not check all n elements, we may miss the desired element.

An even more drastic speedup is attained for the problem of representing a given integer as a product of prime numbers:

- with quantum computing, we can do it in feasible time,

- while for non-quantum computing the only available algorithms require computation time that grows exponentially with the number's length and thus, for 100-digit numbers becomes larger than the lifetime of the Universe.

This example is very important, because all modern computer encryption algorithms – that make our communications private – are based on the difficulty of finding prime factors. So, when quantum computers will appear, all our supposedly secret messages will be available to everyone.

3   From Qubits to Qudits

To describe what are qubits and qudits, let us recall the necessary facts from quantum physics; for details, see, e.g., [7, 22].

States in quantum physics. A specific feature of quantum physics is that for every set of classical states s, ..., s' – which in quantum physics are denoted as $|s\rangle$, ..., $|s'\rangle$ – we can have *superpositions* of these states, i.e., states of the form

$$c_s|s\rangle + \ldots + c_{s'}|s'\rangle,$$

where c_s, ..., $c_{s'}$ are complex numbers for which $|c_s|^2 + \ldots + |c_{s'}|^2 = 1$.

Independent systems in quantum physics. If we have two independent objects, the first one with classical states s, ..., s', and the second one with the states t, ..., t', and the first one is in the state $c_s|s\rangle + \ldots + c_{s'}|s'\rangle$, while the second one is in the state $c'_t|t\rangle + \ldots + c'_{t'}|t'\rangle$, then the system composed of these two objects is in the state

$$c_s \cdot c'_t|s,t\rangle + \ldots + c_s \cdot c'_{t'}|s,t'\rangle + \ldots + c_{s'} \cdot c'_t|s',t\rangle + \ldots + c_{s'} \cdot c'_{t'}|s',t'\rangle.$$

This joint state is called a *tensor product* of the states of these two systems.

Measurements in quantum physics. In general, if in the superposition state $c_s|s\rangle + \ldots + c_{s'}|s'\rangle$, we measure the state of the system, we will get $|s\rangle$ with probability $|c_s|^2$, ..., and $|s'\rangle$ with probability $|c_{s'}|^2$.

The fact that we always get exactly one of these possible results implies that the sum of these probabilities should be equal to 1 – which explains the above condition on the coefficient c_s, ..., $c_{s'}$.

Enter qubits. Most computers are based on the binary system, its components are 2-state components corresponding to binary (0 or 1) digits known as *bits*. So naturally, most current quantum computing schemes are based on using quantum analogues of bits, known as *qubits*.

In particular, since a bit has two states 0 and 1, a general state of a qubit is the state $c_0|0\rangle + c_1|1\rangle$, where c_0 and c_1 are complex numbers for which

$$|c_0|^2 + |c_1|^2 = 1.$$

Traditional approach to implementing qubits. Because of the emphasis on qubits, to implement quantum computing, researchers find quantum systems that can be in several different classical states – e.g., ions – and select one of these states as 0 and another one as 1.

In this usual design of quantum computers, all other classical states – and ions and other physical quantum systems can be in many possible classical states – are not used.

Enter qudits. To further increase efficiency, a natural idea is thus to utilize these additional states. Namely, if we have d different states – which we can mark as states 0, 1, ..., $d-1$ – then a general quantum state of this system has the form

$$c_0|0\rangle + c_1|1\rangle + \ldots + c_{d-1}|d-1\rangle,$$

where $|c_0|^2 + |c_1|^2 + \ldots + |c_{d-1}|^2 = 1$. The states 0, 1, ..., $d-1$ can be naturally labeled by the d-base digits. Because of this labeling, the corresponding quantum states are called *quantum d-base digits*, or *qudits*, for short.

It has been shown that the use of qudits can indeed further speed up quantum computations; see, e.g., [2, 3, 8, 16].

4 How Qudits Can Be Used to Represent Different Types of Uncertainty

Need to represent uncertainty by qudits. Input to computations comes from measurements and expert estimates. In both cases, the values we submit to algorithms are known with uncertainty. It is therefore desirable to represent the corresponding uncertainty in the form appropriate for quantum computing – i.e., by using qudits – or at least by using their particular case of qubits.

In this section, we will show that many types of uncertainty information can indeed be naturally represented in this form.

Case of probabilistic uncertainty. Let us start with the most traditional type of uncertainty – *probabilistic* uncertainty; see, e.g., [21]. In general, such an uncertainty means that we have several (n) alternatives – which we can denote by 0, 1, ..., $n-1$ – and for each alternative i, we know its probability p_i. These probabilities should add up to 1: $p_0 + p_1 + \ldots + p_{n-1} = 1$. A natural qudit representation of this uncertainty means using a qudit with $d = n$ and taking $c_i = \sqrt{p_i}$. For these coefficients, the condition $|c_0|^2 + |c_1|^2 + \ldots = 1$ takes the form $p_0 + p_1 + \ldots = 1$ and is, thus, automatically satisfied.

If we have such qudit representations

$$c_0|0\rangle + \ldots + c_{n-1}|n-1\rangle$$

and

$$c'_0|0'\rangle + \ldots + c'_{m-1}|(m-1)'\rangle$$

of two independent probabilistic objects, with probabilities $p_0, \ldots, p_{n-1}$, and $q_0, \ldots, q_{m-1}$, then, as one can easily see, the system consisting of these two objects is represented by the tensor product of these two qudit states. Indeed, for $c_i = \sqrt{p_i}$ and $c'_j = \sqrt{q_j}$, the probability $|c_i \cdot c'_j|^2$ of getting the state $|i, j\rangle$ is indeed equal to the independence-based value $p_i \cdot q_j$.

Case of fuzzy uncertainty: first idea. What about *fuzzy* uncertainty (see, e.g., [4, 9, 10, 11, 15, 23])? In the fuzzy case, for each of n alternatives, we have a degree $\mu_i \in [0, 1]$ with the condition that $\max(\mu_0, \mu_1, \ldots, \mu_{n-1}) = 1$. In this case, there seems to be no sequence of numbers that adds to 1. To come up with such a sequence, we can use the fact – many times emphasized by Zadeh – that both fuzzy and probabilistic uncertainty can come from the same set of observations, the only difference is in the normalization:

- in the probabilistic case, we normalize so that the sum is equal to 1, while

- in the fuzzy case, we normalize so that the largest value is equal to 1.

This way, we have a natural way to transform probabilities into fuzzy degrees and vice versa:

- if we know the probabilities p_i, then normalization-to-maximum transforms these values into fuzzy degrees:

$$\mu_i = \frac{p_i}{\max(p_0, p_1, \ldots, p_{n-1})};$$

- similarly, if we know fuzzy degrees μ_i, then normalization-to-sum transforms these values into probabilities:

$$p_i = \frac{\mu_i}{\mu_0 + \mu_1 + \ldots + \mu_{n-1}}.$$

So, a natural way to use qudits to represent fuzzy information $\mu_0, \mu_1, \ldots$ is:

- to transform this information into the probabilistic form, and then

- use the above qudit-based representation of probabilities.

Case of fuzzy uncertainty: alternative idea. An alternative idea is to use the fact that in the traditional fuzzy logic, the degree d_+ to which a statement S is true and the degree d_- to which this statement is false add up to 1.

Thus, we can represent such a pair of degrees by a qubit in which $c_0 = \sqrt{d_+}$ and $c_1 = \sqrt{d_-}$.

Case of Pythagorean fuzzy logic. Such a representation is even more natural in *Pythagorean* fuzzy logic (see, e.g., [6]), where the degrees d_+ and d_- are related by a formuyla $d_+^2 + d_2^1 =$. In this case, we can take $c_0 = d_+$ and $c_1 = d_-$.

Case of intuitionistic fuzzy logic. The alternative idea can also be naturally extended to *intuitionistic* fuzzy degrees (see, e.g., [1]), were $d_+ + d_- \leq 1$, and thus, we have $d_+ + d_- + d_0 = 1$, where $d_0 \stackrel{\text{def}}{=} 1 - d_+ - d_-$ is the degree of indifference. To represent such degrees, it is reasonable to use 3-state qudit states $c_0|0\rangle + c_1|1\rangle + c_2|2\rangle$ with $c_0 = \sqrt{d_+}$, $c_1 = \sqrt{d_-}$, and $c_2 = \sqrt{d_0}$.

Case of picture fuzzy logic. A similar representation is possible to *picture fuzzy degrees* (see, e.g., [5]) in which $d_+ + d_- + d_0 \leq 1$ and thus, $d_+ + d_- + d_0 = d_u = 1$, where $d_u \stackrel{\text{def}}{=} 1 - d_+ - d_- - d_0$ is the additional degree. To represent such degrees, it is reasonable to use 4-state qudit states $c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + c_3|3\rangle$ with $c_0 = \sqrt{d_+}$, $c_1 = \sqrt{d_-}$, $c_2 = \sqrt{d_0}$, and $c_3 = \sqrt{d_u}$.

Case of F-transforms. Finally, a useful notion of *F-transform* (see, e.g., [13, 14, 17, 18, 19, 20]) is based on considering families of membership functions $A_0(x), \ldots, A_{n-1}(x)$ for which, for each x, we have $A_0(x) + \ldots + A_{n-1}(x) = 1$. Thus, for each x, we can describe the values of all n basic membership functions by using an n-state qudit with $c_i = \sqrt{A_i(x)}$.

Important comment. It is important to mention that in all these example, we can decrease the number of needed qudit states in half if, instead of considering only real-valued coefficients c_i – as in the current quantum computing algorithms – we allow general complex-valued coefficients $c_i = a_i + b_i \cdot i$, where $i \stackrel{\text{def}}{=} \sqrt{-1}$. In these terms, since $|a + b \cdot i|^2 = |a|^2 + |b|^2$, the condition that $|c_0|^2 + \ldots + |c_{d-1}|^2 = 1$ takes the form

$$|a_0|^2 + |b_0|^2 + \ldots + |a_{d-1}|^2 + |b_{d-1}|^2 = 1.$$

So, e.g., to represent a general picture degree, it is sufficient to use a 2-state qubit

$$c_0|0\rangle + c_1|1\rangle = (a_0 + b_0 \cdot i)|0\rangle + (a_1 + b_1 \cdot i)|1\rangle$$

with $a_0 = \sqrt{d_+}$, $b_0 = \sqrt{d_-}$, $a_1 = \sqrt{d_0}$, and $b_1 = \sqrt{d_u}$.

6

Acknowledgements

References

[1] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Springer-Verlag, Berlin, Heidelberg, 1999.

[2] J. M. Baker and F. T. Chong, "Emerging technologies for quantum computing", *IEEE Micro*, 2021, Vol. 41, No. 5, pp. 41–47.

[3] J. M. Baker, C. Duckering, and F. T. Chong, "Efficient quantum circuit decompositions via intermediate qudits", *Proceedings of the IEEE 50th International Symposium on Multiple-Valued Logic ISMVL'2020*, Miyazaki, Japan, November 9–11, 2020, pp. 303–308.

[4] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[5] Bui Cong Cuong and V. Kreinovich, "Picture Fuzzy Sets - a new concept for computational intelligence problems", *Proceedings of the Third World Congress on Information and Communication Technologies WICT'2013*, Hanoi, Vietnam, December 15–18, 2013, pp. 1–6.

[6] S. Dick, R. Yager, and O. Yazdanbakhsh, "On Pythagorean and complex fuzzy set operations", *IEEE Transactions on Fuzzy Systems*, 2016, Vol. 24, No. 5, pp. 1009–1021.

[7] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.

[8] P. Gokhale, J. M. Baker, C. Duckering, N. C. Brown, K. R. Brown, and F. T. Chong, "Asymptotic improvements to quantum circuits via qutrits", *Proceedings of the 46th International Symposium on Computer Architecture ISCA'19*, Phoenix, Arizona, June 22–26, 2019, pp. 554–566.

[9] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[10] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.

[11] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.

[12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, U.K., 2000.

[13] V. Novak, I. Perfilieva, M. Holcapek, and V. Kreinovich, "Filtering out high frequencies in time series using F-transform", *Information Sciences*, 2014, Vol. 274, pp. 192–209.

[14] V. Novak, I. Perfilieva, and V. Kreinovich, "F-transform in the analysis of periodic signals", In: M. Inuiguchi, Y. Kusunoki, and M. Seki (eds.), *Proceedings of the 15th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CJS'2012*, Osaka, Japan, September 24–27, 2012.

[15] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[16] A. Pavlidis and E. Floratos, *Arithmetic circuits for multilevel qudits based on quantum Fourier transform*, 2017, arXiv:1707.08834.

[17] I. Perfilieva, "Fuzzy transforms: theory and applications", *Fuzzy Sets and Systems*, 2006, Vol. 157, pp. 993–1023.

[18] I. Perfilieva, "F-transform", In: *Springer Handbook of Computational Intelligence*, Springer Verlag, 2015, pp. 113–130.

[19] I. Perfilieva, M. Danková, and B. Bede, "Towards a higher degree F-transform", *Fuzzy Sets and Systems*, 2011, Vol. 180, No. 1, pp. 3–19.

[20] I. Perfilieva, V. Kreinovich, and V. Novak, "F-transform in view of trend extraction", In: *Proceedings of the 15th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CJS'2012*, Inuiguchi, M.; Kusunoki, Y.; Seki, M. (eds.), Osaka, Japan, September 24–27, 2012.

[21] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.

[22] K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.

[23] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.