Applications of Uncertainty Techniques in Science and Engineering: Algebraic Approach

Jaime Nava

Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968
jenava@miners.utep.edu

1. Objectives of Science and Engineering: Prediction

- Prediction is one of the main objectives of science and engineering.
- *Example:* in Newton's mechanics, we want to predict the positions and velocities of different objects.
- *In Physics:*
  - we usually know the exact equations that describe the objects of interest, and
  - we know how to solve these equations.
- This is the case for Newton's mechanics, for example.
- *In this case*, prediction is a purely mathematical problem - of solving the corresponding equations.
- *In practice:* we have uncertainty.

2. Symmetry: a Fundamental Property of the Physical World

- *One of the main objectives of science:* prediction.
- *Basis for prediction:* we observed *similar* situations in the past, and we expect similar outcomes.
- *In mathematical terms:* similarity corresponds to *symmetry*, and similarity of outcomes – to *invariance.*
- *Example:* we dropped the ball, it fall down.
- *Symmetries:* shift, rotation, etc.
- *In modern physics:* theories are usually formulated in terms of symmetries (not diff. equations).
- *Natural idea:* let us use symmetry to describe uncertainty as well.

3. Outline of the Proposed Work

- Prediction and symmetries in describing systems:
  - chemical applications;
  - applications to geosciences;
  - applications to physics.
- Prediction, symmetries in describing uncertainty:
  - neural networks;
  - Dempster-Shafer approach;
  - I-complexity.
- Model validation.
- Design and control:
  - types of intelligent control (Mamdani, etc.);
  - different operations in intelligent control;
  - best intelligent control in terms of tropical algebras.

4. Basic Symmetries: Scaling and Shift

- *Typical situation:* we deal with the numerical values of a physical quantity.
- Numerical values depend on the *measuring unit.*
- *Scaling:* if we use a new unit which is λ times smaller, numerical values are multiplied by λ: $x \to \lambda \cdot x$.
- *Example:* x meters $= 100 \cdot x$ cm.
- *Another possibility:* change the starting point.
- *Shift:* if we use a new starting point which is s units before, then $x \to x + s$ (example: time).
- Together, scaling and shifts form *linear transformations* $x \to a \cdot x + b$.
- *Invariance:* physical formulas should not depend on the choice of a measuring unit or of a starting point.

5. Basic Nonlinear Symmetries

- Sometimes, a system also has *nonlinear* symmetries.
- If a system is invariant under f and g, then:
  - it is invariant under their composition $f \circ g$, and
  - it is invariant under the inverse transformation f^{-1}.
- In mathematical terms, this means that symmetries form a *group*.
- In practice, at any given moment of time, we can only store and describe finitely many parameters.
- Thus, it is reasonable to restrict ourselves to *finite-dimensional* groups.
- *Question* (N. Wiener): describe all finite-dimensional groups that contain all linear transformations.
- *Answer* (for real numbers): all elements of this group are fractionally-linear $x \to (a \cdot x + b)/(c \cdot x + d)$.

6. Symmetries in Chemistry

- Molecules such as benzene or cubane have the additional property of *discrete symmetry*.
- This means that rotation does not change the chemical properties of a molecule.
- *Example:* Perform a clockwise 60 degree rotation on a benzene.

7. First Application: Neural Networks (Brief Reminder)

- In the traditional (3-layer) neural networks, the input values $x_1, \ldots, x_n$:
  - first go through the non-linear layer of "hidden" neurons, resulting in the values
$$y_i = s_0\left(\sum_{j=1}^{n} w_{ij} \cdot x_j - w_{i0}\right) \quad 1 \le i \le m,$$
  - after which a linear neuron combines the results y_i into the output $y = \sum_{i=1}^{m} W_i \cdot y_i - W_0$.
- Here, W_i and w_{ij} are *weights* selected based on the data, and $s_0(z)$ is a non-linear *activation function*.
- Usually, the "sigmoid" activation function is used:
$$s_0(z) = \frac{1}{1 + \exp(-z)}.$$

8. Training a Neural Network: Reminder

- The weights W_i and w_{ij} are selected so as to fit the data, i.e., that

$$y^{(k)} \approx f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right), \text{ where:}$$

  - $x_1^{(k)}, \ldots, x_n^{(k)}$ $(1 \le k \le N)$ are given values of the inputs, and
  - $y^{(k)}$ are given values of the output.

- One of the problems with the traditional neural networks is that
  - in the process of learning – i.e., in the process of adjusting the values of the weights to fit the data –
  - some of the neurons are duplicated, i.e., we get $w_{ij} = w_{i'j}$ for some $i \ne i'$ and thus, $y_i = y_{i'}$.

- As a result, we do not fully use the learning capacity of a neural network: we could use fewer hidden neurons.

9. Apolloni's Idea

- *Problem* (reminder):
  - in the process of learning – i.e., in the process of adjusting the values of the weights to fit the data –
  - some of the neurons are duplicated, i.e., we get $w_{ij} = w_{i'j}$ for some $i \neq i'$ and thus, $y_i = y_{i'}$.
- To avoid this problem, B. Apolloni et al. suggested that we *orthogonalize* the neurons during training.
- In other words, we make sure that the corresponding functions $y_i(x_1, \ldots, x_n)$ remain orthogonal:

$$\langle y_i, y_j \rangle = \int y_i(x) \cdot y_j(x)\, dx = 0.$$

- Since Apolloni *et al.* idea works well, it is desirable to look for its precise mathematical justification.
- We provide such a justification in terms of symmetries.

10. Symmetries in Neural Networks: Why Symmetries?

- At first glance, the use of symmetries in neural networks may sound somewhat strange.
- Indeed, there are no *explicit* symmetries there.
- However, as we will show, *hidden* symmetries have been actively used in neural networks.
- For example, symmetries explain the empirically observed advantages of the sigmoid activation function

$$s_0(z) = \frac{1}{1+\exp(-z)}.$$

11. Symmetries Explain the Choice of an Activation Function

- *What needs explaining:* formula for the *activation function* $f(x) = 1/(1 + e^{-x})$.
- A change in the input starting point: $x \to x + s$.
- *Reasonable requirement:* the new output $f(x+s)$ equivalent to the $f(x)$ mod. appropriate transformation.
- *Reminder:* all appropriate transformations are fractionally linear.
- *Conclusion:* $f(x + s) = \dfrac{a(s) \cdot f(x) + b(s)}{c(s) \cdot f(x) + d(s)}$.
- Differentiating both sides by s and equating s to 0, we get a differential equation for $f(x)$.
- Its known solution is the sigmoid activation function – which can thus be explained by symmetries.

12. Towards Formulating the Problem in Precise Terms

- We select a basis $e_0(x), e_1(x), \ldots, e_n(x), \ldots$ so that each f-n $f(x)$ is represented as $f(x) = \sum_i c_i \cdot e_i(x)$; e.g.:
  - Taylor series: $e_0(x) = 1$, $e_1(x) = x$, $e_2(x) = x^2$, $\ldots$
  - Fourier transform: $e_i(x) = \sin(\omega_i \cdot x)$.
- We store c_0, c_1, $\ldots$, instead of the original f-n $f(x)$.
- *Criterion:* e.g., smallest # of bits to store $f(x)$ with given accuracy.
- *Observation:* storing c_i and $-c_i$ takes the same space.
- Thus, changing one of $e_i(x)$ to $e_i'(x) = -e_i(x)$ does not change accuracy or storage space, so:
  - if $e_0(x), \ldots, e_{i-1}(x), e_i(x), e_{i+1}(x), \ldots$ is an opt. base,
  - $e_0(x), \ldots, e_{i-1}(x), -e_i(x), e_{i+1}(x), \ldots$ is also optimal.

13. Uniqueness of the Optimal Solution

- *Reminder:* we select the basis $\pm e_0(x)$, $\pm e_1(x)$, ...
- Each function is determined modulo its sign.
- Sometimes, we have several optimal solutions.
- Then, we can use an additional criterion; e.g.:
  - if two sorting algorithms are equally fast in the worst case $t^w(A) = t^w(A')$,
  - we can select the one with the smallest average time $t^a(A) \to \min$.
- In effect, we have a new criterion: A is better than A' if $t^w(A) < t^w(A')$ or $(t^w(A) = t^w(A')$ and $t^a(A) < t^a(A'))$.
- So, non-uniqueness means that the original criterion was not final.
- Relative to a *final* criterion, there is *only one* optimal solution.

14. Uniqueness of the Optimal Basis

- *Reminder:*
  - we select the basis $\pm e_0(x)$, $\pm e_1(x)$, $\pm e_3(x)$, $\ldots$;
  - each function is determined modulo its sign.
- *Optimal solutions* are unique:
  - relative to a *final* criterion,
  - there is *only one* optimal solution.
- *Conclusion:* it is reasonable to require that
  - once we have one optimal basis

$$e_0(x), \quad e_1(x), \quad e_2(x), \quad \ldots,$$

  - all other optimal bases have the form

$$\pm e_0(x), \quad \pm e_1(x), \quad \pm e_2(x), \quad \ldots$$

15. How to Describe Average Accuracy

- What is a probability distribution on $f(x)$?
- Dependencies $f(x)$ come from many different factors.
- Due to Central Limit Theorem, it is thus reasonable to assume that the distribution on $f(x)$ is Gaussian.
- If $m(x) \stackrel{\text{def}}{=} E[f(x)] \neq 0$, we can store differences $\Delta f(x) \stackrel{\text{def}}{=} f(x) - m(x)$, for which $E[\Delta f(x)] = 0$.
- Thus, w.l.o.g., we can assume that $E[f(x)] = 0$.
- Such Gaussian distributions are uniquely determined by their covariances $C(x, y) \stackrel{\text{def}}{=} E[f(x) \cdot f(y)]$.
- A Gaussian distribution can be described by indep. components: $f(x) = \sum_i \eta_i \cdot f_i(x)$, w/$E[\eta_i \cdot \eta_j] = 0$, $i \neq j$.
- We also want to know the mean square values $\int (f(x) - f_{\approx}(x))^2 \, dx$.

16. Kahrunen-Loeve (KL) Basis

- A Gaussian distribution can be described by indep. components: $f(x) = \sum_i \eta_i \cdot f_i(x)$, w/$E[\eta_i \cdot \eta_j] = 0$, $i \neq j$.
- We also want to know $\int (f(x) - f_{\approx}(x))^2 \, dx$.
- Idea: use a basis $f_j(x)$ of eigenfunctions of the covariance function $C(x, y) = E[f(x)f(y)]$:

$$\int C(x, y) \cdot f_j(y) \, dy = \lambda_j \cdot f_j(x).$$

- Functions from this *KL basis* are orthogonal; they are usually selected to be orthonormal $\int f_j^2(x) \, dx = 1$.
- If we change some $f_j(x)$ to $-f_j(x)$, we get a KL basis.
- So, criteria depending on $E[f(x) \cdot f(y)]$ and $\int f^2(x) \, dx$ do not change.
- In the general case, when all λ_j are different, each $f_j(x)$ is determined uniquely modulo $f_j(x) \to -f_j(x)$.

17. Proof of the Main Result

- *Let:* $e_i(x)$ be an optimal basis, and let $f_j(x)$ be a KL basis, then $e_i(x) = \sum_j a_{ij} \cdot f_j(x)$.
- *Reminder:* if we change one of the functions $f_{j_0}(x)$ to $-f_{j_0}(x)$, the criterion does not change.
- *Thus:* the following f-ns also form an optimal basis:

$$e_i'(x) = \sum_{j \neq j_0} a_{ij} \cdot f_j(x) - a_{ij_0} \cdot f_{j_0}(x).$$

- *Reminder:* $\forall$ optimal basis has the form $\pm e_i(x)$, thus:

$$e_i'(x) = \sum_{j \neq j_0} a_{ij} \cdot f_j(x) - a_{ij_0} \cdot f_{j_0}(x) = \pm \left(\sum_j a_{ij} \cdot f_j(x) \right).$$

- *So:* if $a_{ij_0} \neq 0$, then $a_{ij} = 0$ for all $j \neq j_0$.
- *Thus:* each $e_i(x)$ has the form $e_i(x) = a_{ij_0} \cdot f_{j_0}(x)$ for some j_0.

18. Conclusions

- *We proved:* that for the optimal basis $e_i(x)$ and for the KL basis $f_j(x)$, each $e_i(x)$ has the form

$$e_i(x) = a_{ij_0} \cdot f_{j_0}(x) \text{ for some } a_{ij_0}.$$

- *We know:* that the elements $f_j(x)$ of the KL basis are orthogonal.
- *So:* we conclude that the elements $e_i(x)$ of the optimal basis are orthogonal as well.
- *Conclusion:* the elements of the optimal basis are orthogonal.
- *Apolloni's idea:* always make sure that we use an orthogonal basis.
- *Fact:* this idea has been empirically successful.
- *New result:* Apolloni's idea has been theoretically justified.

19. Second Application: Kolmogorov Complexity

- The best way to describe the complexity of a given string s is to find its *Kolmogorov complexity* $K(s)$.
- $K(s)$ is the shortest length of a program that computes s.
- For example, a sequence is random if and only if its Kolmogorov complexity is close to its length.
- We can check how close are two DNA sequences s and s' by comparing $K(ss')$ with $K(s) + K(s')$:
  - if they are *unrelated*, the only way to generate ss' is to generate s and then generate s', so

$$K(ss') \approx K(s) + K(s');$$

  - if they are *related*, we have $K(ss') \ll K(s) + K(s')$.

20. Need for Approximate Complexity

- The big problem is that the Kolmogorov complexity is, in general, *not* algorithmically *computable*.
- Thus, it is desirable to come up with *computable* approximations.
- At present, most algorithms for approximating $K(s)$:
  - use some loss-less compression technique to compress s, and
  - take the length $\widetilde{K}(s)$ of the compression as the desired approximation.
- However, this approximation has limitations: for example,
  - in contrast to $K(s)$, where a change (one-bit) change in s cannot change $K(s)$ much,
  - a small change in s can lead to a drastic change in $\widetilde{K}(s)$.

21. I-Complexity

- Limitation of $\widetilde{K}(s)$: a small change in $s = (s_1 s_2 \ldots s_n)$ can lead to a drastic change in $\widetilde{K}(s)$.
- To overcome this limitation, V. Becher and P. A. Heiber proposed the following new notion of *I-complexity*.
- For each position i, we find the length $B_s[i]$ of the largest repeated substring within $s_1 \ldots s_i$.
- For example, for *aaaab*, the corresponding values of $B_s(i)$ are 01233.
- We then define $I(s) \stackrel{\text{def}}{=} \sum_{i=1}^{n} f(B_s[i])$, for an appropriate decreasing function $f(x)$.
- Specifically, it turned out that the *discrete derivative of the logarithm* works well: $f(x) = \text{dlog}(x+1)$, where

$$\text{dlog}(x) \stackrel{\text{def}}{=} \log(x+1) - \log(x).$$

22. Good Properties of I-Complexity

- *Reminder:* $I(s) = \sum_{i=1}^{n} f(B_s[i])$, where:
  - $B_s[i]$ is the length of the largest repeated substring within $s_1 \ldots s_i$, and
  - $f(x) = \log(x+1) - \log(x)$.
- *Similarly to* $K(s)$:
  - If s starts s', then $I(s) \le I(s')$.
  - We have $I(0s) \approx I(s)$ and $I(1s) \approx I(s)$.
  - We have $I(ss') \le I(s) + I(s')$.
  - Most strings have high I-complexity.
- *In contrast to* $K(s)$*:* I-complexity can be computed in linear time.
- *A natural question*: why this function $f(x)$?

23. Towards Precise Formulation of the Problem

- We view the desired function $f(x)$ as a discrete analogue of an appropriate continuous function $F(x)$:

$$f(x) = \int_x^{x+1} g(y)\,dy = F(x+1) - F(x).$$

- Which function $F(x)$ should we choose?
- In the continuous case, the numerical value of each quantity depends:
  - on the choice of the measuring unit and
  - on the choice of the starting point.
- By changing them, we get a new value $x' = a \cdot x + b$.
- For length x, the starting point 0 is fixed.
- So, we only have re-scaling $x \to x' = a \cdot x$.

24. Our Result

- By changing a measuring unit, we get $x' = a \cdot x$.
- When we thus re-scale x, the value $y = F(x)$ changes, to $y' = F(a \cdot x)$.
- It is reasonable to require that the value y' represent the same quantity.
- So, we require that y' differs from y by a similar re-scaling:

  $y' = F(a \cdot x) = A(a) \cdot F(x) + B(a)$ for some $A(a)$ and $B(a)$.
- It turns out that all monotonic solutions of this equation are linearly equivalent to $\log(x)$ or to x^α, i.e.:

$$F(x) = \widetilde{a} \cdot \ln(x) + \widetilde{b} \text{ or } F(x) = \widetilde{a} \cdot x^\alpha + \widetilde{b}.$$

- So, symmetries do explain the selection of the function $F(x)$ for I-complexity.

25. Proof

- *Reminder:* for some monotonic function $F(x)$, for every a, there exist values $A(a)$ and $B(a)$ for which

$$F(a \cdot x) = A(a) \cdot F(x) + B(a).$$

- *Known fact:* every monotonic function is almost everywhere differentiable.
- Let $x_0 > 0$ be a point where the function $F(x)$ is differentiable.
- Then, for every x, by taking $a = x/x_0$, we conclude that $F(x)$ is differentiable at this point x as well.
- For any $x_1 \neq x_2$, we have $F(a \cdot x_1) = A(a) \cdot F(x_1) + B(a)$ and $F(a \cdot x_2) = A(a) \cdot F(x_2) + B(a)$.
- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$.

26. Proof (cont-d)

- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$:

$$F(a \cdot x_1) = A(a) \cdot F(x_1) + B(a).$$
$$F(a \cdot x_2) = A(a) \cdot F(x_2) + B(a).$$

- Thus, both $A(a)$ and $B(a)$ are linear combinations of differentiable functions $F(a \cdot x_1)$ and $F(a \cdot x_2)$.
- Hence, both functions $A(a)$ and $B(a)$ are differentiable.
- So, $F(a \cdot x) = A(a) \cdot F(x) + B(a)$ for differentiable functions $F(x)$, $A(a)$, and $B(a)$.
- Differentiating both sides by a, we get

$$x \cdot F'(a \cdot x) = A'(a) \cdot F(x) + B'(a).$$

- In particular, for $a = 1$, we get $x \cdot \dfrac{dF}{dx} = A \cdot F + B$, where $A \stackrel{\text{def}}{=} A'(1)$ and $B \stackrel{\text{def}}{=} B'(1)$.

27. Proof (final part)

- *Reminder:* $x \cdot \dfrac{dF}{dx} = A \cdot F + B$.
- So, $\dfrac{dF}{A \cdot F + b} = \dfrac{dx}{x}$; now, we can integrate both sides.
- *When $A = 0$:* we get $\dfrac{F(x)}{b} = \ln(x) + C$, so
$$F(x) = b \cdot \ln(x) + b \cdot C.$$
- *When $A \neq 0$:* for $\widetilde{F} \stackrel{\text{def}}{=} F + \dfrac{b}{A}$, we get $\dfrac{d\widetilde{F}}{A \cdot \widetilde{F}} = \dfrac{dx}{x}$, so $\dfrac{1}{A} \cdot \ln(\widetilde{F}(x)) = \ln(x) + C$, and $\ln(\widetilde{F}(x)) = A \cdot \ln(x) + A \cdot C$.
- Thus, $\widetilde{F}(x) = C_1 \cdot x^A$, where $C_1 \stackrel{\text{def}}{=} \exp(A \cdot C)$.
- Hence, $F(x) = \widetilde{F}(x) - \dfrac{b}{A} = C_1 \cdot x^A - \dfrac{b}{A}$.
- The theorem is proven.

28. Third Application: Intelligent Control

- One of the main objectives of fuzzy logic is to formalize commonsense and expert reasoning.
- People use logical connectives like "and" and "or".
- Commonsense "or" can mean both "inclusive or" and "exclusive or".
- *Example:* A vending machine can produce either a coke or a diet coke, but not both.
- In mathematics and computer science, "inclusive or" is the one most frequently used as a basic operation.
- *Fact:* "Exclusive or" is also used in commonsense and expert reasoning.
- *Thus:* There is a practical need for a fuzzy version.
- *Comment:* "exclusive or" is actively used in computer design and in quantum computing algorithms

29. A Crisp "Exclusive Or" Operation

- *Fuzzy analogue* of a classical logic operation op:
  - we know the experts' degree of belief $a = d(A)$ and $b = d(B)$ in statements A and B;
  - based on a and b, we want to estimate the degree of belief in "A op B", as $f_{\text{op}}(a, b)$.
- For op $= \,\&\,$, we get an "and"-operation (t-norm).
- For op $= \vee$, we get an "or"-operation (t-conorm).
- As usual, the fuzzy "exclusive or" operation must be an extension of the corresponding crisp operation $\oplus$.
- In the traditional 2-valued logic, $0 \oplus 0 = 1 \oplus 1 = 0$ and $0 \oplus 1 = 1 \oplus 0 = 1$.
- Thus, the desired fuzzy "exclusive or" operation $f_{\oplus}(a, b)$ must satisfy the same properties:

$$f_{\oplus}(0, 0) = f_{\oplus}(1, 1) = 0; \quad f_{\oplus}(0, 1) = f_{\oplus}(1, 0) = 1.$$

30. Need for the Least Sensitivity: Reminder

- One of the main ways to elicit degree of certainty d is to ask to pick a value on a scale. Example:
  - on a scale of 0 to 10, an expert picks 8, so we get
$$d = 8/10 = 0.8;$$
  - on a scale from 0 to 8, whatever we pick, we cannot get 0.8: $6/8 = 0.75 < 0.8$; $7/8 = 0.875 > 0.8$.
  - the expert will probably pick 6, with
$$d' = 6/8 = 0.75 \approx 0.8.$$
- *It is desirable:* that the result of the fuzzy operation not change much if we slightly change the inputs:
$$|f(a,b) - f(a',b')| \le k \cdot \max(|a-a'|, |b-b'|),$$
with the smallest possible k.
- Such operations are called *the least sensitive* or *the most robust*.

31. For t-Norms and t-Conorms, the Least Sensitivity Requirement Leads to Reasonable Operations

- *Known results:*
  - There is only one least sensitive t-norm ("and"-operation)
$$f_{\&}(a, b) = \min(a, b).$$
  - There is also only one least sensitive t-conorm ("or"-operation)
$$f_{\vee}(a, b) = \max(a, b).$$
- *What we do in this presentation:* we describe the least sensitive fuzzy "exclusive or" operation.

32. Definition of a Fuzzy Exclusive-Or Operation

- **Definition:** *A function* $f : [0,1] \times [0,1] \to [0,1]$ *is called a* fuzzy "exclusive or" operation *if*

$$f(0,0) = f(1,1) = 0 \text{ and } f(0,1) = f(1,0) = 1.$$

- *Comment:* We could also require other conditions, e.g., commutativity and associativity.
- However, our main objective is to select a single operation which is the least sensitive.
- *Fact:* The weaker the condition, the larger the class of operations that satisfy these conditions.
- *Thus:* the stronger the result that our operation is the least sensitive in this class.
- *Conclusion:* We select the weakest possible condition to make our result as strong as possible.

33. Main Result

Definition:

- *Let F be a class of functions from $[0,1]\times[0,1]$ to $[0,1]$.*
- *We say that a function $f \in F$ is* the least sensitive *in the class F if it satisfies the following two conditions:*
  - *for some real number k, the function f satisfies the condition*

$$|f(a,b) - f(a',b')| \le k \cdot \max(|a-a'|, |b-b'|);$$

  - *no other function $f \in F$ satisfies this condition.*

Theorem: *In the class of all fuzzy "exclusive or" operations, the following function is the least sensitive:*

$$f_{\oplus}(a,b) = \min(\max(a,b), \max(1-a, 1-b)).$$

34. Interpretation of the Main Result

- *Reminder:* the least sensitive operation is

$$f_{\oplus}(a,b) = \min(\max(a,b), \max(1-a, 1-b)).$$

- *Fact:* in 2-valued logic, "exclusive or" $\oplus$ can be described in terms of the "inclusive or" operation $\vee$ as

$$a \oplus b \Leftrightarrow (a \vee b)\,\&\neg(a\,\&\,b).$$

- *Natural idea:*
  - replace $\vee$ with the least sensitive "or"-operation $f_{\vee}(a,b) = \max(a,b)$,
  - replace & with the least sensitive "and"-operation $f_{\&}(a,b) = \min(a,b)$, and
  - replace $\neg$ with the least sensitive negation operation $f_{\neg}(a) = 1-a$,
- *Result:* we get the expression given in the Theorem.

35. Interpretation in Terms of Symmetries

- *Reminder:* degrees of certainty do not have a precise numerical meaning, what is important is order.
- *Symmetries:* arbitrary order-preserving transformations $T : [0, 1] \to [0, 1]$.
- *What we need:* "and" and "or" operations $f(a, b)$ which are invariant w.r.t. these symmetries:

$$\text{if } c = f(a, b) \text{ then } T(c) = f(T(a), T(b)).$$

- *If:* $c = f(a, b) \neq a$ and $c \neq b$, then we can have a symmetry T that leaves a and b intact but changes c.
- *In this case:* we have $c = f(a, b)$ but $T(c) \neq f(a, b) = f(T(a), T(b))$.
- *Conclusion:* for a symmetric operation, we should have $f(a, b) = a$ or $f(a, b) = b$, i.e.,

$$f(a, b) = \min(a, b) \text{ or } f(a, b) = \max(a, b).$$

36. Proof of the Main Result: 1st Condition

- *Reminder:* $f_\oplus(a,b) = \min(\max(a,b), \max(1-a, 1-b))$.
- *We need to prove* the following two conditions:
  - *1st:* that this function $f_\oplus(a,b)$ satisfies the following condition with $k = 1$:

$$|f(a,b) - f(a',b')| \le k \cdot \max(|a-a'|, |b-b'|);$$

  - *2nd:* that no other "exclusive or" operation satisfies this property.
- *1st condition:* let us prove that for every $\varepsilon > 0$, if $|a-a'| \le \varepsilon$ and $|b-b'| \le \varepsilon$, then

$$|f_\oplus(a,b) - f_\oplus(a',b')| \le \varepsilon.$$

- *It is known:* that the functions $\min(a,b)$, $\max(a,b)$, and $1-a$ satisfy the above condition with $k = 1$.

37. Proof of the Main Result (cont-d)

- *Known results:* if $|a - a'| \le \varepsilon$ and $|b - b'| \le \varepsilon$, then the following three inequalities hold:

$$|\max(a, b) - \max(a', b')| \le \varepsilon;$$

$$|(1 - a) - (1 - a')| \le \varepsilon; \text{ and } |(1 - b) - (1 - b')| \le \varepsilon.$$

- From the result above, by using the condition for the max operation, we conclude that

$$|\max(1 - a, 1 - b) - \max(1 - a', 1 - b')| \le \varepsilon.$$

- Now, from the results above, by using the condition for the min operation, we conclude that

$$|\min(\max(a, b), \max(1 - a, 1 - b)) -$$

$$\min(\max(a', b'), \max(1 - a', 1 - b'))| \le \varepsilon.$$

- The statement is proven.

38. Fuzzy "Exclusive Or" Operations $f(a,b)$ Which Are the Least Sensitive on Average

- *Idea:* select f so that *on average*, the change in a and b leads to the smallest possible change Δc in $c = f(a,b)$.
- *Assumption:* Δa and Δb are independent random variables with 0 mean and small variance σ^2.
- *Objective:* estimate $\Delta c = f(a + \Delta a, b + \Delta b) - f(a,b)$.
- Since Δa and Δb are small, we can keep only linear terms in the Taylor series of Δc w.r.t. Δa and Δb:

$$\Delta c \approx \frac{\partial f}{\partial a} \cdot \Delta a + \frac{\partial f}{\partial b} \cdot \Delta b.$$

- Since the variables are independent with 0 mean, the mean of Δc is also 0, and variance of Δc is equal to

$$\sigma^2(a,b) = \left(\left(\frac{\partial f}{\partial a} \right)^2 + \left(\frac{\partial f}{\partial b} \right)^2 \right) \cdot \sigma^2.$$

39. Fuzzy "Exclusive Or" Operations Which Are the Least Sensitive on Average (cont-d)

- *Reminder:* for each a and b, the variance $\sigma^2(a,b)$ of Δc is equal to

$$\sigma^2(a,b) = \left(\left(\frac{\partial f}{\partial a}\right)^2 + \left(\frac{\partial f}{\partial b}\right)^2\right) \cdot \sigma^2.$$

- To get the "average" variance, it is reasonable to average this value $\sigma^2(a,b)$ over all possible a and b.
- *Resulting average value:* $I \cdot \sigma^2$, where

$$I \stackrel{\text{def}}{=} \int_{a=0}^{a=1} \int_{b=0}^{b=1} \left(\left(\frac{\partial f}{\partial a}\right)^2 + \left(\frac{\partial f}{\partial b}\right)^2\right) da\, db.$$

- *We want:* the average sensitivity to be the smallest.
- *Conclusion:* we select the function $f(a,b)$ for which the integral I takes the smallest possible value.

40. New Result: Formulation

- *Reminder:* we consider *"exclusive or"* operations $f(a,b)$, i.e., functions $f:[0,1]\times[0,1]\to[0,1]$ for which:

$$f(0,b)=b,\ \ f(a,0)=a,\ \ f(1,b)=1-b,\ \text{and}\ f(a,1)=1-a.$$

- *Main result:* among all such operations, the operation which is the least sensitive on average has the form

$$f_{\oplus}(a,b)=a+b-2\cdot a\cdot b.$$

- *Interpretation:*
  - the classical (2-valued) "exclusive or" operation $a\oplus b$ can be represented as $(a\vee b)\,\&\,(\neg a\vee\neg b)$;
  - use the fuzzy analogues of $\&$, $\vee$, and $\neg$ which are the least sensitive on average:

$$f_{\&}(a,b)=\max(p+q-1,0);\ \ f_{\vee}(a,b)=p+q-p\cdot q;$$

$$f_{\neg}(a)=1-a.$$