Invariance Explains Empirical Success of Many Intelligent Techniques

Olga Kosheleva[1] and Vladik Kreinovich[2]

[1]Department of Teacher Education
[2]Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA
olgak@utep.edu, vladik@utep.edu

1. Need to find dependencies

- One of the main objectives of science is to predict the future state of the world.
- One of the main objectives of engineering is to find out what can be done to make this future state better.
- The state of the world is characterized by the values of relevant quantities.
- For example:
  - to predict tomorrow's weather in a given area,
  - we need to predict temperature, humidity, wind speed, wind direction, and other characteristics in different locations.
- To predict the future state of the world, we can use the current values of these and related quantities.

2. Need to find dependencies (cont-d)

- So, to make a successful prediction, we need to know how exactly the future value of each quantity depends on these current values.
- In computational terms, what we need to know is an algorithm.
- In mathematical terms, what we need to know are functions describing such dependencies.
- Similarly, our possible actions can also be characterized by different numerical parameters.
- To make the future state better, we need to know how the values of the future state depend on these parameters.
- In such problems, we also need to know a function.

3. Need for intelligent techniques

- In many physics situations and in many problems in other application areas, we know the desired function.
- For example, in celestial mechanics, we know exactly how to predict the future locations and velocities of celestial bodies:
  - based on their current locations and velocities – and
  - based on our knowledge about the masses of these bodies.
- However, in many other situations, we do not know the exact dependencies.
- In such situations, at best, we have some partial imprecise knowledge.
- This knowledge needs to be handled by various intelligent techniques such as fuzzy, neural, etc.

4. Need for empirical selection of appropriate functions

- Intelligent techniques provide many different options.
- Many of these options require selecting a function – e.g., selecting an activation function for a neural network.
- Usually in a problem, only some of possible options lead to a success.
- Which option is more successful depends on the situation.
- So, to be successful, we need to empirically select the best option – and thus, the best function.
- In this paper, we summarize the results of empirically selecting best functions in different intelligent techniques.
- We show that in many such cases:
  - the empirical success of the selected function
  - can be explained by their invariance with respect to natural transformations.

5. Need for empirical selection of functions (cont-d)

- These examples include techniques on all level of abstraction:
  - techniques for basic sub-stages of intelligent techniques, e.g., for aggregation and averaging of data points;
  - techniques specific for large segments of the corresponding intelligent techniques, e.g.:
    - the use of non-linear functions in Takagi-Sugeno-type fuzzy control and
    - the use of pooling and averaging in deep learning;
  - intelligent techniques for analyzing specific objects:
    - 1D time series (e.g., related to public transportation),
    - 2D and 3D images (geographical and medical), etc.

6. Need for empirical selection of functions (cont-d)

- From the mathematical viewpoint, our examples will fall into the following three categories:
  - in some cases, we need to select a single function;
  - in other cases, we need to select a family of functions; and
  - in yet other cases, we need to select the best aggregation operation that combines several quantities and/or several functions.
- Of course, the ubiquity of invariances does not mean that:
  - all intelligent techniques
  - can be directly explained by the invariances described in this talk.
- For example, formulas for the normal distribution, the most frequently used type of probabilistic uncertainty cannot be deduced this way.

7. Invariance: shift and scaling

- We want to work with the actual values of physical quantities.
- What we actually work with are numerical values of these quantities.
- This is not just a linguistic distinction.
- The numerical value of a quantity depends not only on its actual values, it also depends on our selection of a measuring unit.
- The same height of 1.7 m gets a different numerical value 170 when described in centimeters.
- In general:
  - if we switch from the original measuring unit to a new unit which is λ times smaller,
  - then all numerical values of the corresponding quantity get multiplied by λ: $x \mapsto \lambda \cdot x$.
- This transformation is known as it *scaling*.

8. Invariance: shift and scaling (cont-d)

- For many quantities like time and (macro-level) temperature, the numerical value also depends on our selection of the starting point.
- In general:
  - if we select a new starting point which is x_0 units smaller than the previous one,
  - then this value x_0 is added to all the numerical values: $x \mapsto x + x_0$.
- This transformation is known as *shift*.

9. Invariance

- In many practical situations, there is no preferred measuring unit and/or no preferred starting point.
- In such situations, it makes sense to require that the corresponding dependencies not depend on these choices.
- For example, suppose that we are interested in the dependence $y = f(x)$ between two quantities x and y.
- Suppose also that there is no preferred measuring unit for measuring x.
- Then it is reasonable to require that the dependence look the same:
  - if we use a different measuring unit and, thus,
  - we use different numerical values of the quantity x, namely, the values $X = \lambda \cdot x$.
- Of course, we cannot simply require that the formula $y = f(x)$ remains the same.

10. Invariance (cont-d)

- That would mean that $y = f(x) = f(X) = f(\lambda \cdot x)$ for all x and λ, which would imply that $f(x)$ is a constant.
- This may sound like a problem at first glance.
- However, e.g.:
  - the formula $d = v \cdot t_0$ – that described how far a body with velocity v can travel during time t_0
  - does not depend on what unit we use to describe velocity.
- But:
  - if we change a measuring unit for velocity, e.g., from km/h to miles per hour,
  - we need to also appropriately change the measuring unit for distance.

11. Invariance (cont-d)

- Thus, we should require that for every $\lambda > 0$, there should be an appropriate transformation $y \mapsto Y$ for which

$$y = f(x) \text{ always implies } Y = f(X).$$

- This is what is usually meant by *invariance*.
- In this talk, we will show how invariance explains many empirical choices.

12. General case

- In general, if we change both the measuring unit and the starting point, we get a generic linear transformation $x \mapsto \lambda \cdot x + x_0$.
- So, in the ideal case, we should be looking for dependencies $f(x)$ which are invariant with respect to both types of transformations.
- Such invariance means that for every $\lambda > 0$ and x_0, there exist values $\mu > 0$ and y_0 depending on λ and x_0, so that:
  - every time we have $y = f(x)$,
  - we also have $Y = f(X)$ for the same function f, where

$$X = \lambda \cdot x + x_0 \text{ and } Y = \mu(\lambda, x_0) \cdot y + y_0(\lambda, x_0).$$

- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $\mu(\lambda, x_0) \cdot y + y_0(\lambda, x_0) = f(\lambda \cdot x + x_0)$.
- Here, $y = f(x)$, so we get $f(\lambda \cdot x + x_0) = \mu(\lambda, x_0) \cdot f(x) + y_0(\lambda, x_0)$.
- Unfortunately, every measurable solution of this functional equation is a linear function.

13. General case (cont-d)

- We want to be able to describe non-linear dependencies – since many real-life dependencies are non-linear.
- Thus, we cannot require invariance with respect to both scaling and shift.
- We can only require one type of invariance.
- Let us see what we can conclude if we make such requirements.

14. Possible cases

- For each of the quantities x and y, we have two possible classes of transformations: scalings and shifts.
- Thus, we can have four possible cases:
  - the case when a scaling of x leads to an appropriate scaling of y;
  - the case when a scaling of x leads to an appropriate shift of y;
  - the case when a shift of x leads to an appropriate scaling of y; and
  - the case when a shift of x leads to an appropriate shift of y.
- Let us consider these four cases one by one.
- After that, we will show, on several examples, that invariance explains the empirical success of the corresponding intelligent techniques.

15. Scaling-scaling case

- In this case, for every $\lambda > 0$, there exists a corresponding value $\mu > 0$ depending on λ so that:
  - every time we have $y = f(x)$,
  - we also have $Y = f(X)$ for the same function f, where $X = \lambda \cdot x$ and $Y = \mu(\lambda) \cdot y$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $\mu(\lambda) \cdot y = f(\lambda \cdot x)$.
- Here, $y = f(x)$, so we get $f(\lambda \cdot x) = \mu(\lambda) \cdot f(x)$.
- It is known that every differentiable solution of this functional equation has the form $f(x) = A \cdot x^a$ for some real numbers A and a.
- Thus, in this case, we get the power law.

16. Comments

- What if we do not require differentiability, we only require that the function $f(x)$ is measurable?
- Then we can have different coefficient A for positive x and for negative x.
- A good example of such a not-everywhere-differentiable function is the Rectified Linear (ReLU) activation function $f(x) = \max(0, x)$.
- This function is used in deep neural networks.
- A similar formula can be obtained in the multi-dimensional case, when the desired quantity y depends on several variables $x_1, \ldots, x_n$.
- In this case, we can select a new measuring unit for each of the n inputs, leading to $x_i \to \lambda_i \cdot x_i$.
- Thus, the requirement that the dependence not depend on the selection of measuring units means the following.

17. Comments (cont-d)

- For every tuple of the values $\lambda_1 > 0, \ldots, \lambda_n > 0$, there exists a value $\mu > 0$ depending on $\lambda_1, \ldots, \lambda_n$ so that:
  - every time we have $y = f(x_1, \ldots, x_n)$,
  - we also have $Y = f(X_1, \ldots, X_n)$ for the same function f, where $X_i = \lambda_i \cdot x_i$ and $Y = \mu(\lambda_1, \ldots, \lambda_n) \cdot y$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $\mu(\lambda_1, \ldots, \lambda_n) \cdot y = f(\lambda_1 \cdot x_1, \ldots, \lambda_n \cdot x_n)$.
- Here, $y = f(x_1, \ldots, x_n)$, so we get

$$f(\lambda_1 \cdot x_1, \ldots, \lambda_n \cdot x_n) = \mu(\lambda_1, \ldots, \lambda_n) \cdot f(x_1, \ldots, x_n).$$

- It is known that every differentiable solution of this functional equation has the form

$$f(x_1, \ldots, x_n) = A \cdot x_1^{a_1} \cdot \ldots \cdot x_n^{a_n} \text{ for some real numbers } A, a_1, \ldots, a_n.$$

18. Scaling-shift case

- In this case, for every $\lambda > 0$, there exists a corresponding value y_0 depending on λ so that:
  - every time we have $y = f(x)$,
  - we also have $Y = f(X)$ for the same function f, where $X = \lambda \cdot x$ and $Y = y + y_0$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $y + y_0(\lambda) = f(\lambda \cdot x)$.
- Here, $y = f(x)$, so we get $f(\lambda \cdot x) = f(x) + y_0(\lambda)$.
- It is known that every diferentiable solution of this functional equation has the form $f(x) = A \cdot \ln(x) + a$ for some real numbers A and a.
- Thus, in this case, we get a logarithmic dependence.

19. Shift-scaling case

- In this case, for every x_0, there exists a corresponding value $\mu > 0$ depending on x_0 so that:
  - every time we have $y = f(x)$,
  - we also have $Y = f(X)$ for the same function f, where $X = x + x_0$ and $Y = \mu(x_0) \cdot y$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $\mu(x_0) \cdot y = f(x + x_0)$.
- Here, $y = f(x)$, so we get $f(x + x_0) = \mu(x_0) \cdot f(x)$.
- It is known that every measurable solution of this functional equation has the form $f(x) = A \cdot \exp(a \cdot x)$ for some real numbers A and a.
- Thus, in this case, we get an exponential dependence.

20. Shift-shift case

- In this case, for every x_0, there exists a corresponding value y_0 depending on x_0 so that:
  - every time we have $y = f(x)$,
  - we also have $Y = f(X)$ for the same function f, where $X = x + x_0$ and $Y = y + y_0(x_0)$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $y + y_0(x_0) = f(x + x_0)$.
- Here, $y = f(x)$, so we get $f(x + x_0) = f(x) + y_0(x_0)$.
- It is known that every measurable solution of this functional equation has the form $f(x) = A \cdot x + a$ for some real numbers A and a.
- Thus, in this case, we get a linear dependence.

21. Comment

- A similar formula can be obtained in the multi-dimensional case, when the desired quantity y depends on several variables $x_1, \ldots, x_n$.
- In this case, we can select a starting point for each of the n inputs, leading to $x_i \mapsto x_i + x_{0i}$.
- Thus, the requirement that the dependence not depend on the selection of the starting point means that:
  - for every tuple of the values $x_{01}, \ldots, x_{0n}$,
  - there exists a corresponding value y_0 depending on $x_{01}, \ldots, x_{0n}$ so that:
    - every time we have $y = f(x_1, \ldots, x_n)$,
    - we also have $Y = f(X_1, \ldots, X_n)$ for the same function f, where $X_i = x_i + x_{0i}$ and $Y = y + y_0(x_{01}, \ldots, x_{0n})$.
- Substituting the expressions for X and Y into the formula $Y = f(X)$, we get $y + y_0(x_{01}, \ldots, x_{0n}) = f(x_1 + x_{01}, \ldots, x_n + x_{0n})$.

22. Comment (cont-d)

- Here, $y = f(x_1, \ldots, x_n)$, so we get

$$f(x_1 + x_{01}, \ldots, x_n + x_{0n}) = f(x_1, \ldots, x_n) + y_0(x_{01}, \ldots, x_{0n}).$$

- It is known that every measurable solution of this functional equation is a linear function

$$f(x_1, \ldots, x_n) = A + a_1 \cdot x_1 + \ldots + a_n \cdot x_n, \text{ for some real numbers } A, a_1, \ldots, a_n.$$

23. Applications of these results

- Intelligent techniques process data.
- This can be 1D data – e.g., several values of measuring the same quantity.
- It can be 2D (or even 3D) data corresponding to images.
- Let's show that the above invariance results can explain empirical successes of intelligent techniques in processing both types of data.

24. Invariance explains empirical successes of intelligent techniques for processing 1D data

- There are many effective techniques for data processing, when:
  - we have the results $x_1, \ldots, x_n$ of measuring or estimating several quantities, and
  - we need to estimate the values of related quantities

$$y = f(x_1, \ldots, x_n),$$

  - e.g., predicting tomorrow's weather based on today's meteorological data (and on the historical meteorological data).
- Somewhat surprisingly, the existing techniques are not as good in a seemingly much simpler 1D problem, when:
  - we have several measurements and/or estimates $x_1, \ldots, x_n$ of the same quantity x, and
  - we would like to "average" them, i.e., to combine them into a single estimate.

25. Invariance explains empirical successes of intelligent techniques for processing 1D data (cont-d)

- In practice, we usually assume that the estimation errors are independent and normally distributed.
- The normal distribution assumption is justified by the Central Limit Theorem, according to which:
  - the joint effect of many small independent factors – which is usually the case
  - is close to normal.
- If we do not have any information about which estimate is more accurate, then it is natural to assume that all estimates are equally accurate.
- In this case, the optimal resulting estimate is the arithmetic average:

$$\overline{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i.$$

26. Invariance explains empirical successes of intelligent techniques for processing 1D data (cont-d)

- In other cases, we know the accuracy of each estimate, i.e., in precise terms, we know the corresponding standard deviations σ_i.
- In such cases, the optimal estimate has the form

$$\overline{x} = \sum_{i=1}^{n} w_i \cdot x_i \text{ for some weights } w_i \geq 0 \text{ for which } \sum_{i=1}^{n} w_i = 1.$$

- The optimal weights depend on the standard deviations σ_i:

$$w_i = \frac{\sigma_i^{-2}}{\sum\limits_{j=1}^{n} \sigma_j^{-2}}.$$

- The problem is that for many quantities, we have many different scales, and the average depends on what scale we use.
- For example, we can describe the strength of an earthquake by the released energy, or we can describe it by the logarithm of this energy.

27. Invariance explains empirical successes of intelligent techniques for processing 1D data (cont-d)

- So, instead of averaging the original values x_i, we can average the transformed values $z_i = g(x_i)$, for some non-linear function $g(x)$:

$$\overline{z} = \sum_{i=1}^{n} w_i \cdot z_i = \sum_{i=1}^{n} w_i \cdot g(x_i).$$

- Then, we re-scale it back into the original scale, resulting in

$$\overline{x} = g^{-1}\left(\sum_{i=1}^{n} w_i \cdot g(x_i)\right),$$ where $g^{-1}(z)$ denotes the inverse function.

- Which function $g(x)$ should be used to get the most accurate results?
- It is reasonable to require that the transformation $g(x)$ is invariant.

28. Invariance explains empirical successes of intelligent techniques for processing 1D data (cont-d)

- Depending on which type of invariance we assume, we get:
  - either power law,
  - or logarithmic dependence,
  - or exponential dependence,
  - or a linear function.
- For linear functions, the above expression leads back to arithmetic average.

29. Invariance explains empirical successes of intelligent techniques for processing 1D data (cont-d)

- For other cases, we get different averaging operations; namely:
  - power-law $g(x)$ leads to $\overline{x} = \left(\sum_{i=1}^n w_i \cdot x_i^a\right)^{1/a}$,
  - logarithmic dependence $g(x)$ leads to $\overline{x} = \prod_{i=1}^n x_i^{w_i}$, and
  - the exponential dependence leads to

$$\overline{x} = \frac{1}{a} \cdot \ln\left(\sum_{i=1}^n w_i \cdot \exp(a \cdot x_i)\right).$$

- In the limits $a \to \infty$ and $a \to -\infty$, the power-law formula leads to $\overline{x} = \max(x_1, \ldots, x_n)$ and to $\overline{x} = \min(x_1, \ldots, x_n)$.
- These are exactly the empirically effective averaging operations – so these operations are explained by invariance.

30. Comment

- In some cases, the most effective averaging operations are more complex, e.g., Lehmer means $\overline{x} = \dfrac{\dfrac{1}{n} \cdot \displaystyle\sum_{i=1}^{n} w_i \cdot x_i^a}{\dfrac{1}{n} \displaystyle\sum_{i=1}^{n} w_i \cdot x_i^{a-1}}$.
- This operation can also be explained by scale-invariance – it is the result of applying:
  - a scale-invariant function of two variables
  $$y = f(y_1, y_2) = y_1^a \cdot y_2^{-(a-1)}$$
  - to averaging operations corresponding to scale-invariant functions $g_1(x) = x^a$ and $g_2(x) = x^{a-1}$:
  $$y_1 = \left(\sum_{i=1}^{n} w_i \cdot x_i^a\right)^{1/a} \text{ and } y_2 = \left(\sum_{i=1}^{n} w_i \cdot x_i^{a-1}\right)^{1/(a-1)}.$$

31. Comment (cont-d)

- Other empirically successful operation are the following operations:

$$\overline{x} = p \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} x_i + (1-p) \cdot \min(x_1, \ldots, x_n) \text{ and}$$

$$\overline{x} = p \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} x_i + (1-p) \cdot \max(x_1, \ldots, x_n) \text{ for some } p \in [0,1].$$

- These operations are obtained if we:
  - apply a shift-invariant combination operation – i.e., linear combination
  - to two invariant averaging operations: average and min (or max).

32. Invariance explains empirical successes of intelligent techniques for processing 2D and 3D data

- Current image processing techniques have achieved great performance in image processing, e.g:
  - in detecting objects in images,
  - in describing different types of well-defined relationship between these objects.
- The current methods are, however, not as successful in describing intuitive, imprecise relationship between objects.
- We humans are accustomed to describe relative position of objects in imprecise terms: close by, far away, somewhat to the East, etc.
- We use these descriptions to make decisions.

33. Invariance explains empirical successes of intelligent techniques for processing 2D and 3D data (cont-d)

- It is therefore necessary to be able:
  - given a scene,
  - to generate an appropriate description of relative positions of different objects in such understandable natural-language terms.
- For this task, one of the most efficient methods is a Histogram of Forces method, in which:
  - for each direction,
  - we integrate the force $F(r)$ over all the lines parallel to this direction.
- Here r is the shortest distance between points from these two objects that happen to be on this particular line.
- The effectiveness of this method depends on the choice of $F(r)$.

34. Invariance explains empirical successes of intelligent techniques for processing 2D and 3D data (cont-d)

- Empirically, it turned out that the most effective functions are the power laws $F(r) = A \cdot r^a$.
- The most widely used cases are $a = 0$ (constant force) and $a = -2$.
- The expression for $a = -2$ is the same as for the gravitational force between two bodies.
- Other values of a have also shown to be effective in some cases, e.g., the value $a = 2$.
- The power laws are exactly the functions which are scale-scale invariant.
- Since, as we have mentioned, scale-scale-invariance is a natural property, this explains the empirical success of these functions.

35. Comment

- In some cases, other functions $F(r)$ turned out to be the most effective, e.g., the functions $F(r) = \min(C, r^{-2})$ and $F(r) = r_0^2/(r_0^2 + r^2)$.
- In the following slides, we will show that the general invariance ideas explain the effectiveness of such functions as well.

36. Why we need families of functions

- So far, we were looking for a single function that would be most efficient in solving several problems.
- Often, in different applications, different functions are more effective.
- In this case, instead of looking for a *single* function, it makes sense to look for a finite-parametric *family* of functions.
- We would then be able to adjust the values of the corresponding parameters for each individual case.
- Let us analyze which families are invariant.

37. What families we will consider

- The simplest families are linear combinations of known functions, i.e., families of the type $C_1 \cdot f_1(x) + \ldots + C_k \cdot f_k(x)$.
- Here the functions $f_1(x), \ldots, f_k(x)$ are fixed, and $C_1, \ldots, C_k$ are the parameters that can be adjusted.

38. Which families are scale-invariant and shift-invariant

- For a single function, we could not require invariance with respect to both changing the measuring unit and changing the starting point.
- If we require both, we only get linear functions.
- Interestingly, for families, it *is* possible to require both invariances.
- It turns out that the only family of functions which is both scale-invariant and shift-invariant is the family of polynomials:

$$f(x) = C_1 + C_2 \cdot x + C_3 \cdot x^2 + \ldots + C_k \cdot x^{k-1}.$$

- A similar results holds if we consider scale-invariant and shift-invariant families of functions of several variables.
- All elements of such functions are polynomials.

39. Which families are scale- and/or shift-invariant (cont-d)

- If we require only shift-invariance, then:
  - all functions from the invariant family are linear combinations of the expressions $x^m \cdot \exp(a \cdot x)$,
  - where m is non-negative integer and a is, in general, a complex number.
- If we require only scale-invariance, then:
  - all functions from the invariant family are linear combinations of the expressions $(\ln(x))^m \cdot x^a$,
  - where also m is non-negative integer and a is, in general, a complex number.
- For this result, we will describe two applications:
  - a general application to intelligent systems and
  - a more specific application to human-oriented systems.

40. General application to intelligent systems

- As we have mentioned, in many situations of prediction and control:
  - we do not have exact knowledge of the system that we want to control, but
  - we have knowledge of experts who have been successfully predicting and/or successfully controlling such systems.
- The problem is that experts formulate their knowledge in imprecise ("fuzzy") terms, by using imprecise words from natural language.
- To reformulate such a knowledge in precise computer-understandable terms, Zadeh invented fuzzy techniques.
- One of the most successful ways to use this technique in control is to use Takagi-Sugeno approach.

41. General application to intelligent systems (cont-d)

- To describe how experts predict or control based on the inputs $x_1, \ldots, x_n$, we look for expert rules of the type

$$\text{if } A_1(x_1), \ldots, \text{ and } A_n(x_n), \text{ then } y = f(x_1, \ldots, x_n).$$

- Here, A_i are fuzzy properties and $f(x_1, \ldots, x_n)$ is a linear function.
- The pioneering paper (Tanaka 2009) showed that more effective prediction and control can be obtained if we allow polynomial functions $f(x_1, \ldots, x_n)$.
- This approach is known as *polynomial fuzzy approach.*
- We know that invariance is a natural property, and all the functions from scale-invariant and shift-invariant families are polynomials
- So, this explains the empirical success of polynomial fuzzy approach.

42. Application to human-oriented systems

- Intelligent control techniques are largely used in situations when the objective is clear.
- For human-oriented system, an additional challenge is that for such systems, the goal is subjective.
- It deals with perceptions and not with objective characteristics.
- To design such systems, we need to be able to predict such perceptions.
- In other words, we need to know how a human will react to different situations.
- As a case study, let us consider the planning of public transportation – an important feature of big cities and of their smart-city plans.
- The goal of planning is to make the public transportation as convenient to people as possible.
- The main source of inconvenience is waiting time.

43. Application to human-oriented systems (cont-d)

- Part of the waiting time is caused by the fact that the buses (and other means of public transportation) go by schedule.
- So a passenger has to wait for the next scheduled bus.
- This is a known inconvenience.
- People get adjusted to it – by taking the bus schedule into account when planning their trips.
- A much more serious inconvenience occurs when the buses are behind schedule.
- Such situations are unpredictable, they interfere with people's plans and cause a lot of frustration.

44. Application to human-oriented systems (cont-d)

- According to (Tran 2022), there are several levels of such frustration, corresponding to 10, 30, and 60 minutes:
  - delays below 10 min are perceived as negligible,
  - delays between 10 and 30 min are perceived as short,
  - delays between 30 and 60 minutes are perceived as long, and
  - delays longer than 60 minutes are perceived as severe.
- According to decision theory, people's attitudes can be described by a function called utility.
- The larger the utility the more beneficial the situation.
- Utility is defined modulo possible linear transformation $u \mapsto \lambda \cdot u + u_0$ for some $\lambda > 0$ and u_0.
- Very small changes in a situation lead to very small, barely noticeable changes in utility.
- Let us denote by Δu the smallest noticeable change in utility.

45. Application to human-oriented systems (cont-d)

- Let us take, as a starting point for measuring utility, the value corresponding to no delay.
- Then, the first noticeable change 10 min correspond to utility $-\Delta u$, the second (30 min) to $-2\Delta u$, and the third one (60 min) to $-3\Delta u$.
- In general, how can we describe the dependence $t = f(u)$ of delay time t on utility u?
- This dependence may be different for different people.
- So it makes sense to select not a single function $t = f(u)$, but a whole family of functions.
- Since utility is defined modulo scalings and shifts, it is reasonable to require that this family should be scale-invariant and shift-invariant.
- Thus, it must be a family of polynomials.
- The simplest polynomials are linear functions.

46. Application to human-oriented systems (cont-d)

- However, a linear dependence would mean the following.
- A change from 5 min delay to 15 min delay is as painful as a change from 60 min delay to 70 min delay.
- In reality, if we have already waited for the bus for the whole hour, additional 10 minutes are not very painful.
- However, in the first case, the delay time triples.
- To capture this difference – which is not reflected in the linear dependence – we need to go beyond linear functions.
- The simplest family of polynomials that includes non-linear functions is the family of all quadratic polynomials.
- And indeed, quadratic functions perfectly explain the above empirical dependence.

47. Application to human-oriented systems (cont-d)

- Let us select parameters C_i of the quadratic function

$$t(u) = C_1 + C_2 \cdot u + C_3 \cdot u^2 \text{ so that}$$

$$t(0) = 0, \quad t(-\Delta u) = 10, \text{ and } t(-2\Delta u) = 30.$$

- Then, we will get $C_1 = 0$, $C_2 = -\dfrac{5}{\Delta u}$, $C_3 = \dfrac{5}{(\Delta u)^2}$.
- For these values, we get $t(-3\Delta u) = 60$, exactly what the empirical data shows.
- So, in this case, invariance also explains an empirical dependence.
- Interestingly, the dependence of amount of money on the utility is also quadratic.
- The reason is similar: a change from \$5 to \$15 causes much more positive feelings than a change from \$60 to \$70.

48. What is an aggregation operation

- In many practical situations, we need to combine (aggregate) two or more values.
- For example, when we know the masses m_1 and m_2 of two objects, their total mass m is equal to the sum of their masses: $m = m_1 + m_2$.
- Similarly:
  - when we have two independent random variables with known standard deviations σ_1 and σ_2,
  - then the standard deviation σ of their sum is equal to $\sqrt{\sigma_1^2 + \sigma_2^2}$.
- The function that transforms the original values a_1 and a_2 into a new value is known as an *aggregation operation*.
- We will denote such operations by $a_1 * a_2$.
- The combination result should not depend on the order.
- So it is usually assumed that this operation should be commutative:

$$a_1 * a_2 = a_2 * a_1.$$

49. What is an aggregation operation (cont-d)

- For the same reason, often, associativity is required.
- In most practical situations, it is reasonable to require monotonicity: if $a_1 \le a_1'$ and $a_2 \le a_2'$, then we should have $a_1 * a_2 \le a_1' * a_1'$.

50. Aggregation operations and averaging operations

- Once we have an aggregation operation $a_1 * a_2$, we can define the corresponding averaging operation.
- It transforms the values $a_1, \ldots, a_n$ into the value a for which

$$a_1 * \ldots * a_n = a * \ldots * a \text{ (n times)}.$$

- For example, if we start with the sum $a_1 * a_2 = a_1 + a_2$, then we get arithmetic average $\dfrac{a_1 + \ldots + a_n}{n}$.
- If we start with the product $a_1 * a_2 = a_1 \cdot a_2$ then we get geometric average $\sqrt[n]{a_1 \cdot \ldots \cdot a_n}$, etc.

51. Which aggregation operations are invariant

- Let us first consider scale-invariance, i.e., the property that:
  - if $a = a_1 * a_2$,
  - then, for every $\lambda > 0$, we should have $A = A_1 * A_2$, where $A = \lambda \cdot a$, $A_1 = \lambda \cdot a_1$, and $A_2 = \lambda \cdot a_2$.
- It is known that scale-invariant, commutative, associative, continuous, and monotonic aggregation operations are:
  - the operation $a_1 * a_2 = (a_1^v + a_2^v)^{1/v}$,
  - its limit cases $a_1 * a_2 = \max(a_1, a_2)$ and $a_1 * a_2 = \min(a_1, a_2)$ corresponding to $v \to \infty$ and $v \to -\infty$, and
  - the trivial operation for which $a_1 * a_2 = 0$ for all a_1 and a_2.
- We can also require shift-invariance, i.e., require that $a = a_1 * a_2$ imply that $A = A_1 * A_2$, where $A = a + a_0$, $A_1 = a_1 + a_0$, and $A_2 = a_2 + a_0$.
- Then the only remaining non-trivial aggregation operations are min and max.

52. Which aggregation operations are invariant (cont-d)

- We can require the following weaker version of shift-invariance.
- $a = a_1 * a_1$ implies that $A = A_1 * A_2$ for $A_1 = a_1 + a_0$, $A_2 = a_2 + a_0$, and $A = a + a_0'(a_0)$ for some value a_0' depending on a_0.
- Then we also get addition – which, as we have mentioned, corresponds to the arithmetic average.

53. Applications

- As promised, we will show that the above results explain effectiveness of empirical intelligent techniques on all levels of abstraction:
  - on the level of general techniques – in this case, it will be deep learning,
  - on the level of specific applications – in this case, it will be applications to imaging, and
  - on the level of building blocks – in this case, it will be aggregation.

54. Applications to deep learning

- One of the main ideas behind deep learning – as compared to few-layers traditional neural networks – is that:
  - we have more layers and,
  - correspondingly, fewer neurons in each layer.
- In the traditional neural network, we had a large number of neurons in each layer – in particular, in the input layer.
- So, we could allocated, to each input value, a corresponding input neuron.
- Often, we process a large amount of data – e.g., pixels forming an image.
- In this case, there are much fewer neurons in the input layer than there are inputs.
- So, we need to combine several inputs into a single value.
- This process is known as *pooling*.

55. Applications to deep learning (cont-d)

- It is reasonable to require that this operation be scale-invariant and – at least weakly – shift-invariant.
- So, it is reasonable to use max-pooling, min-pooling, or average pooling.
- These are indeed three most widely used pooling operations in deep learning.
- Thus, the empirical success of these operations can also be explained by invariance.
- Another problem of machine learning in general – and of deep learning in particular – is that its results are not always reliable.
- In general, a natural way to increase the reliability is to duplicate efforts, i.e., to have several similar devices working in parallel.
- Then, we somehow average their results.

56. Applications to deep learning (cont-d)

- This is, e.g., how we get the most accurate time:
  - by having three or more super-precise clocks working in parallel and
  - by averaging their results.
- Similarly, to increase reliability of a neural network:
  - we simultaneously train several networks – i.e., in effect, subnetworks of the overall network, and
  - then we average the results.
- Thus, it makes sense to use averaging corresponding to invariant aggregations, e.g., arithmetic average.
- This is also one of the two most empirically effective averaging methods.
- The other empirically effective method is geometric average that corresponds to $v \to 0$.

57. First application to image processing

- Now it is time to go to the image processing example.
- As we have mentioned earlier, scale-invariance leads to the power law
$$F(r) = A \cdot r^a.$$
- In some cases, some values a are better, in other cases, other values of a are better.
- It is therefore reasonable to try to aggregate functions corresponding to different values a.
- The hope is that the resulting function combine the advantages of both aggregated expressions.
- By applying a scale-invariant aggregation, we get one of the following expressions:
$$((A_1 \cdot r^{a_1})^v + (A_2 \cdot r^{a_2})^v)^{1/v}, \quad \max(A_1 \cdot r^{a_1}, A_2 \cdot r^{a_2}),$$
$$\min(A_1 \cdot r^{a_1}, A_2 \cdot r^{a_2}).$$

58. First application to image processing (cont-d)

- In particular:
  - if we apply the simplest of such combinations – min – to the most successful cases $a_1 = 0$ and $a_2 = -2$,
  - we get one of the hybrid force formulas $F(r) = \min(C, r^{-2}, C)$ that was empirically shown to be effective.
- For $v = -1$, $A_1 = 1$, $A_2 = r_0^2$, $a_1 = 0$, and $a_2 = -2$, we get another empirically successful formula $F(r) = r_0^2/(r_0^2 + r^2)$.
- Thus, invariance explains these empirical successes as well.

59. Second application to image processing: gauging quality of skin lesion segmentation

- Not only we need to make intelligent techniques more reliable.
- We also need to be able to gauge how reliable they are.
- A recent study (Lin 2022) provides a new method for this gauging, and use it for skin lesion segmentation.
- This new method combines:
  - taking the arithmetic average – of several images provided by different subnetworks, and
  - taking min – namely, the minimum of the distances from each pixel to different points on the boundaries between the segments.
- These are exactly the scale-invariant and (weakly) shift-invariant aggregation operation.
- Thus, invariance explains the empirical success of (at least this part of) the new method.

60. Application to building blocks of intelligent techniques: hierarchical aggregation

- Aggregation does not have to be performed in one step.
- For example, to get an average temperature on campus, it makes sense:
  - first to aggregate temperature values within each room – if the room has several sensors,
  - then combine these values to get a building average, and
  - then aggregate the building averages to get the overall campus average.

61. Application to building blocks of intelligent techniques: hierarchical aggregation (cont-d)

- The study of different aggregation techniques described in (Magdalena 2022) showed that:
  - in line with the above-mentioned result,
  - the best results emerge when on each aggregation stage, we use min, max, or arithmetic average.
- Thus, this empirical result is also explained by invariance.

62. Comment

- The paper (Magdalena 2022) has an additional empirical observation:
  - on the first aggregation levels, it is advantageous to use min and max, while
  - on the following levels, arithmetic average works better.
- This fact can be explained by the fact that with some small probability:
  - sensors malfunction, and
  - produce readings which are much larger or much smaller than the actual temperature.
- On the earlier aggregation stages, we combine the readings of a small number of sensors.
- Thus, the probability that one of the readings is an outlier is still small.

63. Comment (cont-d)

- So, with probability close to 1, the max and min of these readings reflect the actual highest and lowest temperature in the room.
- On the other hand, on the later aggregation stages, we combine, in effect, a large number of readings.
- In this cases, there is a high probability that at least one of the combined values is an outlier; thus:
  - if we simply use max or min to aggregate,
  - then with high probability, we will get this outlier – and not the desired highest or lowest temperature on campus.
- Since we cannot use max or min, the only remaining option is to compute the arithmetic average.

64. Acknowledgments

- This work was supported in part by the National Science Foundation grants:
  - 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and
  - HRD-1834620 and HRD-2034030 (CAHSI Includes).
- It was also supported by the AT&T Fellowship in Information Technology.
- It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.