Noname manuscript No.
(will be inserted by the editor)

Granular Approach to Data Processing Under Probabilistic Uncertainty

Andrzej Pownuk · Vladik Kreinovich

Received: date / Accepted: date

Abstract The existing algorithms for data processing under probabilistic uncertainty often require too much computation time. Sometimes, we can speed up the corresponding computations if we take into account the fact that in many real-life situations, uncertainty can be naturally described as a combination of several components, components which are described by different granules. In such situations, to process this uncertainty, it is often beneficial to take this granularity into account by processing these granules separately and then combining the results. In this paper, we show that granular computing can help even in situations when there is no such natural decomposition into granules: namely, we can often speed up processing of uncertainty if we first (artificially) decompose the original uncertainty into appropriate granules.

Keywords Granular computing · Probabilistic Uncertainty · Faster Algorithm

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

A. Pownuk and V. Kreinovich
Computational Science Program
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
Tel. +1-915-757-6951
Fax: +1-915-747-5030
E-mail: ampownuk@utep.edu, vladik@utep.edu

1 Data Processing under Probabilistic Uncertainty: General Formulation of the Problem

Need for data processing. Often, we are interested in a quantity y which is difficult (or even impossible) to measure or estimate directly. For example, y can be the amount of oil in a given oilfield, distance to a faraway star, or the future value of a quantity of interest.

To estimate this value y, we:

- find easier-to-measure and/or or easier-to-estimate quantities $x_1, \ldots, x_n$ which are related to y by a known dependence $y = f(x_1, \ldots, x_n)$,
- measure x_i's, and
- use the results $\widetilde{x}_i$ of measuring x_i and the known relation $y = f(x_1, \ldots, x_n)$ to estimate y as

$$\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n).$$

Applying the known algorithm f to measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ is an important case of *data processing*.

For example, to find the distance to a faraway star, we can:

- measure the directions x_1 and x_2 to this star in two different seasons, when the Earth is on the opposite sides of the Sun, and then
- use trigonometry to find the desired distance y.

To estimate the amount of oil in a given oil field, we:

- perform a large number of seismic experiments – by setting up small explosions and measuring resulting seismic waves at different locations;
- then, we can use known algorithms for solving the corresponding systems of partial differential equations (that describe wave propagation) to estimate the desired quantity y based on the results $\widetilde{x}_i$ of seismic measurements.

As a simple example of data processing, we can use the application of Ohm's law $V = I \cdot R$ relating the voltage V, the current I, and the resistance R. Because of this law, to measure the voltage, we can:

- measure the current $x_1 = I$ passing through a resistor with known resistance $x_2 = R$, and then
- estimate the desired voltage $y = V$ as $\widetilde{y} = f(\widetilde{x}_1, \widetilde{x}_2)$, where the corresponding algorithm has a simple form $f(x_1, x_2) = x_2 \cdot x_2$.

Comment. In this example, the algorithm is very simple, but, as we have mentioned earlier, in general, we may have very complex algorithms $f(x_1, \ldots, x_n)$ for processing data.

Need to take uncertainty into account. Due to measurement uncertainty, the measurement results $\widetilde{x}_i$ are, in general, different from the actual values x_i of the corresponding quantities. In other words, we have a non-zero *measurement error*

$$\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i.$$

Because of these measurement errors, the value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ that we obtain by processing the measurement results is, in general, different from the desired value $y = f(x_1, \ldots, x_n)$.

For example:

- if the actual value of the current is $x_1 = 1.0$ and the actual value of the resistance is $x_2 = 2.0$, then we should get $y = 1.0 \cdot 2.0 = 2.0$.
- However, if we measure with uncertainty, we may get e.g., $\widetilde{x}_1 = 1.1$ and $\widetilde{x}_2 = 1.9$, in which case the result $\widetilde{y} = \widetilde{x}_1 \cdot \widetilde{x}_2 = 1.1 \cdot 1.9 = 2.09$ will be slightly different from the desired value $y = 2$.

In general, it is important to estimate the resulting uncertainty $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$; see, e.g., (Rabinovich 2005).

For example, if we estimate that the oil field contains approximately 100 million cubic meters, then at first glance, this is good news: since this estimate makes the oil field very rich and worth exploiting. However, due to measurement uncertainty, this value is inaccurate, and what action to take depends on the accuracy:

- If it is 100 ± 10, this is great news.
- However, if it is 100 ± 200, then maybe there is practically no oil in this field at all, so it may be better to perform some further measurements before we invest money in digging an expensive oil well.

Enter probabilistic uncertainty. For measurements, we usually have a large number of situations when we performed the measurement with our measuring instrument and we also measured the same quantity with some more accurate measuring instrument – so that we have a good record of past values of measurement errors. For example, we may record the temperature outside by a reasonably cheap not-very-accurate thermometer, and we can find the measurement errors by comparing these measurement results with accurate measurements performed at a nearby meteorological station.

Based on such a record, we can estimate the probabilities of different values of the measurement error. Thus, it is reasonable to assume that for each i, we know the distribution of the measurement error

In some cases, we have eliminated all major sources of measurement error. As a result, the remaining measurement error is a joint effect of a large number of small difficult-to-eliminate effects. Due to the Central Limit Theorem (see, e.g., (Sheskin 2011)), in this case, the probability distribution of the measurement error is close to Gaussian.

However, such an elimination is a very time-consuming and expensive process, it is usually only performed on expensive super-accurate measuring instruments. In practice, the distributions are often different from Gaussian.

Measurement errors corresponding to different variables are usually independent.

We thus arrive at the following problem.

Problem P: the main problem. *We know:*

- *the dependence $y = f(x_1, \ldots, x_n)$,*
- *the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$, and*
- *the probability distributions of each of the variables $\Delta x_i = \widetilde{x}_i - x_i$.*

We assume that the random variables Δx_i are independent.

We need to find the probability distribution of the quantity

$$\Delta y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n).$$

Comment. The exact formulation of the problem depends on the exact form in which the information about the probability distributions is known, and in what form we want the information about the probability distribution of Δy. The corresponding probability distributions can be represented by the values of their probability density functions $\rho(z)$, by their moments, by the parameters within the appropriate family, etc.

Linear case. The simplest possible functions are linear functions

$$y = f(x_1, \ldots, x_n) = c_0 + c_1 \cdot x_1 + \ldots + c_n \cdot x_n.$$

In this case, subtracting this formula from

$$\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) = c_0 + c_1 \cdot \widetilde{x}_1 + \ldots + c_n \cdot \widetilde{x}_n,$$

we conclude that

$$\Delta y = c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n.$$

For such functions, the above main problem takes the following form:

Problem P^L: linear version of the main problem. *We know:*

- *the values $c_1, \ldots, c_n$, and*
- *the probability distributions of each of the variables $\Delta x_i = \widetilde{x}_i - x_i$.*

We assume that the random variables Δx_i are independent.

We need to compute the probability distribution of the quantity $\Delta y = c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n$.

Measurement errors are usually relatively small. We emphasized the linear case because in covers most practical situations. Indeed, measurement errors are usually relatively small, so that terms quadratic in measurement errors can be safely ignored (Rabinovich 2005).

For example, if the measurement error is 10%, then the square of this value is 1% which is definitely much smaller than 10%.

In this case, we can expand the expression

$$\Delta y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n)$$

in Taylor series and ignore terms which are quadratic (or higher order) in terms of Δx_i. Then, we get

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i, \tag{1}$$

where we denoted $c_i \stackrel{\text{def}}{=} \dfrac{\partial f}{\partial x_i}_{|x_j = \widetilde{x}_j}$.

Example. In the Ohm's law example, when $f(x_1, x_2) = x_1 \cdot x_2$, we have $c_1 = \dfrac{\partial f}{\partial x_1} = x_2$ and $c_2 = \dfrac{\partial f}{\partial x_1} = x_1$, so the above formula takes the form

$$\Delta y = \widetilde{x}_2 \cdot \Delta x_1 + \widetilde{x}_1 \cdot \Delta x_2.$$

We can reduce the general problem to the linearized case. Thus, we can reduce the general problem P to its linearized case P^L;

Algorithm scheme S_L for reducing P to P^L. *Suppose that we know the dependence $y = f(x_1, \ldots, x_n)$, the measurement results $\widetilde{x}_i$, and the probability distributions for each Δx_i. Then, we can find the probability distribution for Δy as follows:*

- *first, we compute the values of the partial derivatives:*

$$c_i = \frac{\partial f}{\partial x_i}_{|x_j = \widetilde{x}_j};$$

- *then, we use an algorithm for solving the linearized problem P^L compute the probability distribution of the quantity*

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i.$$

Comments.

- The partial derivatives can be computed either analytically – by using the general formulas of differentiation – or by using numerical differentiation

$$\frac{\partial f}{\partial x_i} \approx \frac{f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}}{h}$$

  for some small h.
- Because the general case of the main problem can be reduced to the linearized case, in the following text, we will concentrate on soling the linearized version P^L of the main problem.

2 Existing General Methods for Solving the Main Problem: Description and Limitations

Existing methods. Because of the practical importance of data processing under probabilistic uncertainty, there are many techniques for solving this problem, both in the case when the relation is given in the form of an explicit algorithm (see, e.g., (Rabinovich 2005) and references therein), and in the case when this relation is given in terms of systems of equations; see, e.g., (Kovalerchuk and Kreinovich 2017; Kovalerchuk 2016; Piegat and Landowski 2018). These methods are overviewed in this section.

Most of these methods reduce the above problem to its simplest case when all the coefficients c_i are equal to 1. Let us formulate this simplified case in general terms.

Problem P^S: simplified version of the main problem.

- *We know the probability distributions of each of n independent random variables $t_1, \ldots, t_n$.*
- *We want to compute the probability distribution of their sum $t = \sum_{i=1}^{n} t_i$.*

The general linearized problem can be reduced to this simplified case.

Algorithm scheme $S_{L\to S}$ for reducing P^L to P^S. *Suppose that we know the probability distributions for each of the random variables Δx_i and we know the coefficients c_i. Then, we can find the probability distribution for $\Delta y = \sum\limits_{i=1}^{n} c_i \cdot \Delta x_i$ as follows:*

- *first, for each i from 1 to n, we compute the probability distribution for the quantity $t_i = c_i \cdot \Delta x_i$;*
- *then, we use an algorithm for solving the simplified problem P^S to find the probability distribution of the sum $\Delta y = \sum\limits_{i=1}^{n} t_i$.*

Comments.

- For most representations of the probability distribution, transforming the distribution for Δx_i to distribution for $t_i = c_i \cdot \Delta x_i$ is easy. For example:
  - if we know the pdf $\rho_{\Delta x}(\Delta x_i)$, then the corresponding pdf for t_i has the form
$$\rho_t(t_i) = \frac{1}{|c_i|} \cdot \rho_{\Delta x}\left(\frac{t_i}{c_i}\right);$$
  - if we know the mean μ_i and the variance V_i of the random variable Δx_i, then the mean μ_i' and V_i' of the variable $t_i = c_i \cdot \Delta x_i$ has the form
$$\mu_i' = c_i \cdot \mu_i \text{ and } V_i' = c_i^2 \cdot V_i.$$
- Since the main problem M can be reduced to its linearized case P^L, and this case, in its turn, can be reduced to the simplified case P^S, in the following text, we will mostly concentrate on soling the simplified version P^S of the main problem.

Reduction to the case when $n = 2$. We can simplify the problem even further, by reducing is to the case of $n = 2$. For $n = 2$, the simplified case takes the following form.

Problem $P^{S(n=2)}$: Case $n = 2$ of the simplified problem P^S.

- *We know the probability distributions of each of two independent random variables t_1 and t_2.*
- *We want to compute the probability distribution of their sum $t = t_1 + t_2$.*

Once we know how to solve this simplest case $n = 2$ of the general problem, we can then solve the original more general problem in the following way:

Algorithm scheme $S_{S\to S(n=2)}$ for reducing P^S to $P^{S(n=2)}$.

- *First, we apply the $n = 2$ algorithm to find the probability distribution of the sum $s_2 \stackrel{\text{def}}{=} t_1 + t_2$.*
- *Then, we again apply the $n = 2$ algorithm, this time to the random variables s_2 and t_3, to find the probability distribution of their sum*
$$s_3 \stackrel{\text{def}}{=} s_2 + t_3 = (t_1 + t_2) + t_3 = \sum_{i=1}^{3} t_i.$$
- *After that, we apply the $n = 2$ algorithm to the random variables s_3 and t_4, to find the probability distribution of their sum*
$$s_4 \stackrel{\text{def}}{=} s_3 + t_4 = \sum_{i=1}^{3} t_i + t_4 = \sum_{i=1}^{4} t_i,$$
etc.

Comment. At the end, we get the desired probability distribution for the sum
$$\Delta y = \sum_{i=1}^{n} t_i \left(= \sum_{i=1}^{n} c_i \cdot \Delta x_i. \right)$$

So, to solve our original problem of data processing under probabilistic uncertainty, it is sufficient to be able to solve the $n = 2$ version $P^{S(n=2)}$ of the simplified problem P^S.

What if know the pdfs. As we have mentioned earlier, the exact formulation of the main problem – and thus, of its linearized and simplified versions – depends on how we represent the original probability distributions and in what form we want to have the resulting probability distribution.

Usually, a probability distribution is described by its probability density function (pdf) $\rho(z)$. A natural way to represent a general function in a computer is by the values of this function on a grid $z_i = z_0 + i \cdot h$ for some step h, where $i = 0, 1, \ldots, N$ for some N. For this representation, we arrive at the following version of the problem P^S:

Problem P^S_ρ: case when we know the pdfs. *We know that $t = t_1 + \ldots + t_n$ and that the random variables t_i are independent.*

- *For each i from 1 to n, we know the values $\rho_i(t_{ij})$ of the pdf $\rho_i(z)$ of the variable t_i for $t_{1j} = t_{10} + j \cdot h$.*
- *We want to compute the values $\rho(v_k)$ of the sum's pdf on the grid $v_k = v_0 + k \cdot h$, where*
$$v_0 = t_{10} + \ldots + t_{n0}.$$

So solve this problem, we can take into account that, e.g., for $n = 2$, the pdf of the sum $t = t_1 + t_2$ of two independent random variables with pdfs $\rho_1(t_1)$ and $\rho_2(t_2)$ is equal to (Sheskin 2011)
$$\rho(t) = \int \rho_1(t_1) \cdot \rho_2(t - t_1)\, dt_1.$$

To find the desired values $\rho(v_k)$, we can therefore approximate the above integral by the integral sum, and get

$$\rho(v_k) = \sum_{i=0}^{k} \rho_1(t_{1i}) \cdot \rho_2(t_{2,k-i}) \cdot h. \qquad (2)$$

This leads to the following known straightforward algorithm.

Straightforward algorithm $A_\rho^{S(n=2)}$ for solving the problem $P_\rho^{S(n=2)}$. *Suppose that we know the values $\rho_i(t_{ij})$ of the original probability distributions $\rho_i(t)$ for the values $t_{ij} = t_{i0} + j \cdot h$.*

Then, we can compute the values $\rho(v_k)$ of the desired probability distribution for $t = t_1 + t_2$ for values $v_k = t_{10} + t_{20} + k \cdot h$ by using the formula (2).

The main limitation of the above straightforward approach is that it often requires a large amount of computation time. Indeed, according to the above algorithm, to compute each value $\rho(v_k)$, we need to perform $2k$ elementary computational steps:

- we need k multiplications,
- we need $k-1$ additions, and
- we need one multiplication by h.

Thus, overall, to compute all $2N$ values of $\rho(v_k)$, we need to perform

$$2 + 4 + 6 + \ldots + 2 \cdot (2N) = 2 \cdot (1 + 2 + \ldots + N) =$$

$$2 \cdot \frac{N \cdot (N+1)}{2} = N \cdot (N+1) = O(N^2)$$

computational steps.

If we want to have an accurate representation of the corresponding probability distribution on a given interval $[\underline{t}, \overline{t}]$, i.e., if we want a reasonable small step h, then we need to use a reasonably large number

$$N = \frac{\overline{t} - \underline{t}}{h}.$$

For large N, performing N^2 steps is feasible but rather slow; see, e.g., (Cormen at al. 2009).

Comment. Good news is that there exist faster methods; see below.

What is we know characteristic functions. A known faster algorithm is based on the possibility of describing each probability distribution not by its probability density function $\rho(x)$, but by a *characteristic function*

$$\chi(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot x)] = \int \exp(\mathrm{i} \cdot \omega \cdot x) \cdot \rho(x)\, dx,$$

where $E[\cdot]$ denotes the expected value.

In the computer, a natural idea is to represent each characteristic function $\chi(\omega)$ by its values $\chi(\omega_j)$ on a grid $\omega_j = \omega_0 + j \cdot h$.

In these terms, the above basic problem takes the following form:

Problem P_χ^S: case when we know the characteristic functions. *We know that $t = t_1 + \ldots + t_n$ and that the random variables t_i are independent.*

- *For each i from 1 to n, we know the values $\chi_i(\omega_j)$ of the characteristic function $\chi_i(\omega)$ of the variable t_i for $\omega_j = \omega_0 + j \cdot h$, $j = 0, 1, \ldots$*
- *We want to compute the values $\chi(\omega_j)$ of the sum's characteristic function on the same grid ω_j.*

The reason why the use of characteristic functions leads to faster computations is that, e.g., for $t = t_1 + t_2$, we have

$$\exp(\mathrm{i} \cdot \omega \cdot t) = \exp(\mathrm{i} \cdot \omega \cdot t_1) \cdot \exp(\mathrm{i} \cdot \omega \cdot t_2).$$

Since t_1 and t_2 are independent, we have

$$E[\exp(\mathrm{i} \cdot \omega \cdot t)] = E[\exp(\mathrm{i} \cdot \omega \cdot t_1)] \cdot E[\exp(\mathrm{i} \cdot \omega \cdot t_2)].$$

So, for the corresponding characteristic functions $\chi_i(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot t_i)]$ and $\chi(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot t)]$, we get

$$\chi(\omega) = \chi_1(\omega) \cdot \chi_2(\omega),$$

for all ω. In particular, for each k, we get

$$\chi(\omega_k) = \chi_1(\omega_k) \cdot \chi_2(\omega_k). \qquad (3)$$

Similarly, for the sum $t = t_1 + \ldots + t_n$ of n independent random variables, we get

$$\chi(\omega_k) = \chi_1(\omega_k) \cdot \ldots \cdot \chi_n(\omega_k). \qquad (4)$$

This leads to the following known algorithm:

Algorithm A_χ^S for solving the problem P_χ^S.

- *We start with $N \cdot n$ values $\chi_i(\omega_k)$ describing the characteristic functions of the random variables*

$$t_1, \ldots, t_n.$$

- *Then, for each k, we apply the formula (4) to compute the desired value $\chi(\omega_k)$.*

This algorithm requires $n-1$ multiplications for each of N values $\chi(\omega_k)$. So, overall, this algorithm requires $O(N)$ computational steps – which is, for large N, much faster than the N^2 steps needed for the straightforward Algorithm $A_\rho^{S(n=2)}$.

We can apply the characteristic function-based algorithm to the case when we know the pdfs. The above algorithm for solving the problem P_{χ}^{S} is applicable if the information about the probability distributions comes in terms of the characteristic functions, and we want the resulting distribution presented in the same form.

In many practical situations, however, the probability distributions are given in terms of their pdfs, and it is desirable to get the resulting distribution also in terms of the pdf. In such situations, we can still utilize the above algorithm, but for that, we need to solve the following two auxiliary problems:

- first, we need to transform the information about each input t_i from the pdf to the characteristic function form, and
- second, we need to transform the information about the output t back into the pdf form.

Let us describe this in precise terms.

Auxiliary problem $T_{\rho\to\chi}$ of transforming ρ into χ.

- *We know the values $\rho(t_j)$ of the pdf on a grid t_j.*
- *We want to compute the values $\chi(\omega_k)$ of the characteristic function on the corresponding grid.*

Auxiliary problem $T_{\chi\to\rho}$ of transforming χ into ρ.

- *We know the values $\chi(\omega_k)$ of the characteristic function on a grid ω_k.*
- *We want to compute the values $\rho(t_j)$ of the pdf on he corresponding grid t_j.*

The characteristic function is, in effect, a Fourier transform of the pdf. Thus, to solve both auxiliary problems, i.e., to transform the probability density function into a characteristic function and back – we can use the Fast Fourier Transform algorithm; see, e.g., (Cormen at al. 2009).

Thus, we arrive at the following algorithm for solving the pdf version of the main problem.

Characteristic-function-based algorithm $A^{S}_{\rho\,(\text{via }\chi)}$ for solving the pdf-based problem P_{ρ}^{S}. *Suppose that we know the pdfs $\rho_i(t_{ik})$ of n independent random variables $t_1,\ldots,t_n$. Then, to compute the pdf $\rho(t)$ of their sum, we do the following:*

- *first, for each variable t_i, we apply the Fast Fourier transform to the given pdf values $\rho_i(t_{ij})$; as a result, we get the values $\chi_i(\omega_k)$ of the corresponding characteristic function $\chi_i(\omega)$;*
- *then, for each k, we compute the values*

$$\chi(\omega_k)=\prod_{i=1}^{n}\chi_i(\omega_k);$$

- *finally, we apply the Fast Inverse Fourier Transform to the resulting values $\chi(\omega_k)$ and thus get the desired values $\rho(t_j)$ of the pdf for the sum t.*

The Fast Fourier Transform algorithm takes

$$O(N\cdot\log(N))$$

steps. In this case, the overall computation time for solving the problem P_{ρ}^{S} is

$$O(N\cdot\log(N))+O(N)=O(N\cdot\log(N)).$$

For large N, this computation time is much smaller than N^2. Thus:

- even if the original information is given in terms of a probability density function, and
- as a result, we want a probability density function,

the use of the above characteristic-function algorithm is much faster than the straightforward approach.

Remaining problem. When we compare Algorithm A_{ρ}^{S} for solving the pdf-based problem with the algorithm A_{χ}^{S} for solving the characteristic-function-based problem, we see that by changing the representation of a random variable, we achieved a drastic speedup, from $O(N^2)$ to $O(N)$.

The remaining problem is that for large N, performing $O(N)$ steps may still take too long – especially if we perform these computations inside a measuring device, when computational ability is limited (e.g., by limitations on size or on energy consumption). A natural question is: can we find an alternative representation of probability distributions that would lead to even faster solution of our main computational problem?

As of now, no such representation is known in the general case – but, as we will show in the next sections, such faster-to-process representations *are* known for some specific classes of probability distributions.

In this paper, we show that granular computing ideas enable us to extend these results to the general case.

3 Solving the Main Problem for Specific Classes of Probability Distributions: Existing Fast Algorithms

Case of normal distributions. The most well-known case when a faster algorithm for solving the main problem is possible is the case when all distributions Δx_i are normal (Gaussian). In this sum, the linear combination $\Delta y=\sum\limits_{i=1}^{n}c_i\cdot\Delta x_i$ is also normal. A general normal distribution can be described by two parameters: mean μ

and variance V. Thus, we arrive at the following problem:

Problem P_N^L: case of normal distributions.

- *We know the means $\mu_1, \ldots, \mu_n$ and the variances $V_1, \ldots, V_n$ of n independent normally distributed random variables $\Delta x_1, \ldots, \Delta x_n$.*
- *We know the coefficients $c_1, \ldots, c_n$.*
- *We want to compute the mean μ and the variance V of their linear combination $\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i$.*

It is known that the mean of the linear combination of n random variables is equal to the linear combination of their means. We also know that the variance of the variable $t_i = c_i \cdot \Delta x_i$ is equal to $c_i^2 \cdot V_i$, and that for independent random variables t_i, the variance of their sum is equal to the sum of their variances. Thus, we arrive at the following algorithm for solving the above problem:

Algorithm A_N^L for solving the problem P_N^L. *Once we know the values $\mu_1, \ldots, \mu_n$, $V_1, \ldots, V_n$, and $c_1, \ldots, c_n$, we compute $\mu = c_1 \cdot \mu_1 + \ldots + c_n \cdot \mu_n$ and*

$$V = c_1^2 \cdot V_1 + \ldots + c_n^2 \cdot V_n.$$

To compute each sum, we need to perform $n - 1$ additions. Thus, overall, we need to perform $2(n - 1)$ computational steps – which is much smaller than

$$N \cdot (n - 1)$$

computations needed for the algorithm based on characteristic functions.

Comment. Of course, this reduction of computation time is only possible for a specific class of distributions – namely, for the normal ones. **What if the information is given in terms of the pdf.** In the situation when the information about the random variables is given in terms of the pdf and/or we want to information about the sum t in terms of the pdf, we need to perform the corresponding transformations, i.e., we need to solve the following two auxiliary problems.

Auxiliary problem $T_{\rho \to N}$ of transforming ρ into μ and V.

- *We know the values $\rho(t_j)$ of the pdf on a grid t_j.*
- *We need to compute the mean μ and the variance V of the corresponding distribution.*

Auxiliary problem $T_{N \to \rho}$ of transforming μ and V into ρ.

- *We know the mean μ and the variance V of the normal distribution.*
- *We need to compute the values $\rho(t_j)$ of the pdf on a grid t_j.*

The first auxiliary problem can be solved if we use discrete approximations to the usual integral formulas $\mu = \int t \cdot \rho(t)\, dt$ and that $V = \int t^2 \cdot \rho(t)\, dt - \mu^2$, i.e., we compute

$$\mu = \sum_j t_j \cdot \rho(t_j) \cdot h;$$

$$V = \sum_j t_j \cdot \rho(t_j) \cdot h - \mu^2.$$

These computations require time $O(N)$.

The second auxiliary problem can be solved by using the usual formula for the pdf of the normal distribution

$$\rho(t_j) = \frac{1}{\sqrt{2\pi} \cdot \sqrt{V}} \cdot \exp\left(-\frac{(t_j - \mu)^2}{2V}\right),$$

which also requires N computational steps.

So, if the data is given in terms of the pdf and we want the result in terms of the pdf, we need $O(N) + 2 + O(N) = O(N)$ computational steps – much fewer than $O(N \cdot \log(N))$ steps needed when we use the general characteristic-function-based algorithm.

Comment. Of course, this reduction of computation time is only possible for a specific class of distributions – namely, for the normal ones.

A more general case – of infinitely divisible distributions. The above fast algorithm for solving the main problem in the case of normal distributions is based on the fact that the sum of two or more independent normally distributed random variables is also normally distributed. There are other probability distributions with this property.

This same property can be described in a different way: that a normally distributed random variable can be "subdivided" into (i.e., represented as a sum of) many small normally distributed parts. Because of this reformulation, such probability distributions are known as *infinitely divisible*. All classes of infinitely divisible distributions are known, they are described by characteristic functions of the form

$$\chi(\omega) = \exp(\mathrm{i} \cdot \mu \cdot \omega - A \cdot |\omega|^\alpha - B \cdot \mathrm{sign}(\omega) \cdot |\omega|^\alpha).$$

For example:

- For $\alpha = 2$ and $B = 0$, we get normal distributions.

– For $\alpha = 1$ and $B = 0$, we get Cauchy distributions, with the probability density function

$$\rho(x) = \frac{1}{\pi \cdot \Delta} \cdot \frac{1}{1 + \dfrac{(x-\mu)^2}{\Delta^2}}$$

for an appropriate $\Delta > 0$.

It is easy to check that, for each α, similar to normal distributions, a linear combination of random variables distributed according to such distributions, also has a similar distribution.

Indeed, if a random variable x has such a distribution, then, for each real number c, the characteristic function $\chi'(t)$ of the new variable $t = c \cdot x$ has the form

$$\chi'\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot t)] = E[\mathrm{i} \cdot \omega \cdot (c \cdot x)] =$$

$$E[\mathrm{i} \cdot (\omega \cdot c) \cdot x] = \chi(c \cdot \omega).$$

Substituting the above expression for $\chi(\omega)$ into this formula, we conclude that $\chi'(\omega) = \exp(C)$, where

$$C = \mathrm{i} \cdot c \cdot \omega - A \cdot |c|^\alpha \cdot |\omega|^\alpha - B \cdot \mathrm{sign}(c) \cdot |c|^\alpha \cdot \mathrm{sign}(\omega) \cdot |\omega|^\alpha,$$

i.e., has the same form with $\mu' = c \cdot \mu$, $A' = |c|^\alpha \cdot A$, and $B' = \mathrm{sign}(c) \cdot |c|^\alpha \cdot B$.

Similarly, if we have a sum $t = t_1 + \ldots + t_n$ of several independent random variables, with characteristic functions

$$\chi_i(\omega) = \exp(\mathrm{i} \cdot \mu_i \cdot \omega - A_i \cdot |\omega|^\alpha - B_i \cdot \mathrm{sign}(\omega) \cdot |\omega|^\alpha),$$

then the characteristic function $\chi(\omega)$ for the sum t has the form

$$\chi(\omega) = \chi_1(\omega) \cdot \chi_2(\omega) =$$

$$\exp(\mathrm{i} \cdot \mu \cdot \omega - A \cdot |\omega|^\alpha - B \cdot \mathrm{sign}(\omega) \cdot |\omega|^\alpha),$$

where $\mu = \mu_1 + \ldots + \mu_n$, $A = A_1 + \ldots + A_n$, and

$$B = B_1 + \ldots + B_n.$$

This allows us to come up with a method for solving the main problem when each distribution is represented by the parameters of the corresponding infinitely divisible distribution.

Problem P_I^L: case of infinitely divisible distributions. *Let α be fixed.*

– *We know the values $\mu_1, \ldots, \mu_n$, $A_1, \ldots, A_n$, and $B_1, \ldots, B_n$ of n independent random variables*

$$\Delta x_1, \ldots, \Delta x_n$$

distributed according to the corresponding infinitely divisible distributions.
– *We know the coefficients $c_1, \ldots, c_n$.*
– *We want to compute the values μ, A, and B describing the distribution of the quantity $\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i$.*

Algorithm A_I^L for solving the problem P_I^L. *Once we know the values μ_i, A_i, and B_i, we compute*

$$\mu = c_1 \cdot \mu_1 + \ldots + c_n \cdot \mu_n,$$

$$A = |c_1|^\alpha \cdot A_1 + \ldots + |c_n|^\alpha \cdot A_n, \text{ and}$$

$$B = \mathrm{sign}(c_1) \cdot |c_1|^\alpha \cdot B_1 + \ldots + \mathrm{sign}(c_n) \cdot |c_n|^\alpha \cdot B_n.$$

To compute each sum, we need to perform $O(n)$ additions. Thus, overall, we need to perform $O(n)$ computational steps – the amount not depending on N. Thus, for large N, the number of computational steps is much smaller than $N \cdot (n-1)$ computations needed for the algorithm based on characteristic functions.

Of course, this reduction of computation time is only possible for a specific class of distributions – namely, for the infinitely divisible ones. In the following section, we show how to extend this idea to general probability distributions.

What if the information is given in terms of the pdf. In the situation when the information about the random variables is given in terms of the pdf and/or we want to information about the sum t in terms of the pdf, we need to perform the corresponding transformations, i.e., we need to solve the following two auxiliary problems.

Auxiliary problem $T_{\rho \to I}$ of transforming ρ into μ, A, and B.

– *We know the values $\rho(t_j)$ of the pdf on a grid t_j.*
– *We need to compute the parameters μ, A, and B describing the corresponding distribution.*

Auxiliary problem $T_{I \to \rho}$ of transforming μ, A, and B into ρ.

– *We know the values μ, A, and B describing the corresponding distribution.*
– *We need to compute the values $\rho(t_j)$ of the pdf on a grid t_j.*

The first auxiliary problem can be solved if we:

– first perform the Fourier transform and get the characteristics function $\chi(\omega)$;
– then, compute its logarithm.

This logarithm has the form

$$\ln(\chi(\omega)) = \mathrm{i}\cdot\mu\cdot\omega - A\cdot|\omega|^{\alpha} - B\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha}.$$

So, we can find the desired values μ, A, and B by solving the resulting systems of linear equations – e.g., by using the usual Least Squares approach.

Comment. It should be mentioned that a new, more efficient algorithm for solving the first auxiliary problem will be described later in this paper.

To solve the second auxiliary problem, we can:

- use the explicit formula for the characteristic function of an infinitely divisible distribution to compute the values $\chi(\omega_k)$, and then
- use Fourier transform to compute the desired cdf.

4 A New Granular-Based Algorithm for Data Processing Under Probabilistic Uncertainty

One of the main ideas of granular computing: reminder. One of the main ideas of granular computing is to approximate the general difficult-to-to-process information by a combination of easier-to-process units – *granules*; see, e.g., (Pedrycz and Chen 2011, 2015,a).

How we use this general idea. In our case, as we have mentioned:

- easier-to-process units are infinitely divisible distribution, and
- a natural combination rule – that we used throughout the paper – is a linear combination (or, for simplicity, addition) of the corresponding independent random variables.

Thus, a natural idea is to approximate each random variable Δx_i by a sum

$$\Delta x_i = r_{i1} + \ldots + r_{ij} + \ldots + r_{ik}$$

of infinitely divisible random variables r_{ij} corresponding to different values of α_j, μ_{ij}, A_{ij}, and B_{ij}. By combining these functions, we can approximate any possible distribution.

To make combination of such variables easier, it makes sense to select several values $\alpha_1, \ldots, \alpha_k$ – e.g., $\alpha_1 = 1$ and $\alpha_2 = 2$ – and use the same values α_j for all random variables Δx_i.

A reasonable idea is to have these values uniformly distributed on the interval $[1, 2]$, by taking

$$\alpha_j = 1 + \frac{j-1}{k-1}.$$

For example:

- for $k = 2$, we get $\alpha_1 = 1$ and $\alpha_2 = 2$,
- for $k = 3$, we get $\alpha_1 = 1$, $\alpha_2 = 1.5$, and $\alpha_3 = 2$, etc.

Comment. A (slightly) better selection of the values α_j is described in the Appendix.

Once the values α_j are fixed, the characteristic function $\chi_{ij}(\omega)$ of each variable r_{ij} has the form

$$\chi_{ij}(\omega) = \exp(\mathrm{i}\cdot\mu_{ij}\cdot\omega - A_{ij}\cdot|\omega|^{\alpha_j} - B_{ij}\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}).$$

Thus, the characteristic function $\chi_i(\omega)$ of the sum $\Delta x_i = \sum\limits_{j=1}^{k} r_{ik}$ is equal to the product

$$\chi_i(\omega) = \prod_{j=1}^{k} \chi_{ij}(\omega) =$$

$$\exp\left(\mathrm{i}\mu_i\omega - \sum_{j=1}^{k} A_{ij}|\omega|^{\alpha_j} - \sum_{j=1}^{k} B_{ij}\mathrm{sign}(\omega)|\omega|^{\alpha_j}\right),$$

where $\mu_i \stackrel{\text{def}}{=} \sum\limits_{j=1}^{k} \mu_{ij}$.

With this granular representation of the probability distributions, our main problem takes the following form:

Problem P_G^L: case of granular representation. *Let a sequence of values $\alpha_1, \ldots, \alpha_k$ be fixed.*

- *For each i, we know the values μ_i, $A_{i1}, \ldots, A_{ik}$, and $B_{i1}, \ldots, B_{ik}$ describing the probability distribution of each of n independent random variables*

  $$\Delta x_1, \ldots, \Delta x_n$$

  distributed according to the corresponding infinitely divisible distributions.
- *We know the coefficients $c_1, \ldots, c_n$.*
- *We want to compute the values μ, $A_1, \ldots, A_k$, and $B_1, \ldots, B_k$ describing the distribution of the quantity $\Delta y = \sum\limits_{i=1}^{n} c_i \cdot \Delta x_i$.*

How can we solve this problem? Similarly to the previous section, we can show that for each i, the characteristic function $\chi_i'(\omega)$ of the quantity $t_i = c_i \cdot \Delta x_i$ has the form

$$\chi_i'(\omega) = \chi_i(c_i\cdot\omega) =$$

$$\exp\left(\mathrm{i}\cdot c_i\cdot\mu_i\cdot\omega - \sum_{j=1}^{k} A_{ij}\cdot|c_i|^{\alpha_j}\cdot|\omega|^{\alpha_j} - \right.$$

$$\left.\sum_{j=1}^{k} B_{ij}\cdot\mathrm{sign}(c_i)\cdot|c_i|^{\alpha_j}\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}\right).$$

So, we get a representation with $\mu_i' = c_i \cdot \mu_i$, $A_{ij}' = A_{ij} \cdot |c_i|^{\alpha_j}$, and $B_{ij}' = B_{ij} \cdot \mathrm{sign}(c_i) \cdot |c_i|^{\alpha_j}$.

The characteristic function $\chi(\omega)$ of the sum $\Delta y = t_1 + \ldots + t_n$ of these random variables is equal to the product of the corresponding characteristic functions:

$$\chi(\omega) = \chi_1'(\omega) \cdot \ldots \cdot \chi_n'(\omega),$$

and thus, has the form

$$\chi(\omega) =$$

$$\exp\left(\mathrm{i}\mu\omega - \sum_{j=1}^{k} A_j|\omega|^{\alpha_j} - \sum_{j=1}^{k} B_j \mathrm{sign}(\omega)|\omega|^{\alpha_j}\right),$$

where $\mu = \mu_1' + \ldots + \mu_n'$, $A_j = A_{1j}' + \ldots + A_{nj}'$, and

$$B_j = B_{1j}' + \ldots + B_{nj}'.$$

Substituting the expressions for μ_i', A_{ij}', and B_{ij}' into this formula, we arrive at the following algorithm for solving the main problem in the case of granular representation of a probability distribution.

Algorithm A_G^L for solving the problem P_G^L. *We are given:*

- *the values μ_i, A_{i1}, …, A_{ik}, B_{i1}, …, B_{ik} that represent the random variables Δx_i ($i = 1, \ldots, n$); and*
- *the values c_1, …, c_n.*

To find the value μ, A_1, …, A_k, B_1, …, B_k corresponding to $\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i$, we perform the following operations:

- *first, we compute $\mu = \sum_{i=1}^{n} c_i \cdot \mu_i$;*
- *then, for each value j from 1 to k, we compute*

$$A_j = \sum_{i=1}^{n} |c_i|^{\alpha_j} \cdot A_{ji} \text{ and } B_j = \sum_{i=1}^{n} \mathrm{sign}(c_i) \cdot |c_i|^{\alpha_j} \cdot B_{ij}.$$

This algorithm requires $O(k \cdot n)$ computational steps – independently on N. So, for large N, this algorithm is much faster than the algorithm based on characteristic functions.

Comment. It is worth mentioning that a similar idea can be applied to the case of fuzzy uncertainty; see (Stylios, Pownuk, and Kreinovich 2015) for details.

What if the information is given in terms of the pdf. In the situation when the information about the random variables is given in terms of the pdf and/or we want to information about the sum t in terms of the pdf, we need to perform the corresponding transformations, i.e., we need to solve the following two auxiliary problems.

Auxiliary problem $T_{\rho \to G}$ of transforming ρ into μ, A_j, and B_j.

- *We know the values $\rho(t_j)$ of the pdf on a grid t_j.*
- *We need to compute the parameters μ, $A_1, \ldots, A_k$, and $B_1, \ldots, B_k$ describing the corresponding distribution.*

Auxiliary problem $T_{G \to \rho}$ of transforming μ, A_j, and B_j into ρ.

- *We know the values μ, $A_1, \ldots, A_k$, and $B_1, \ldots, B_k$ describing he corresponding distribution.*
- *We need to compute the values $\rho(t_j)$ of the pdf on a grid t_j.*

The solution to the second auxiliary problem is reasonably straightforward:

Algorithm $A_{G \to \rho}$ for transforming μ, A_j, B_j into ρ. *Suppose that we are given the values $\alpha_1, \ldots, \alpha_k$ and the corresponding values μ, $A_1, \ldots, A_k$, and $B_1, \ldots, B_k k$. Then:*

- *first, we form a characteristic function, i.e., we compute the values*

$$\chi(\omega_k) =$$

$$\exp\left(\mathrm{i}\omega_k - \sum_{j=1}^{k} A_j|\omega_k|^{a_j} - \sum_{j=1}^{k} B_j \mathrm{sign}(\omega_k)|\omega_k|^{\alpha_j}\right)$$

  for points $\omega_k = \omega_0 + k \cdot h$ on a grid;
- *then, we apply Inverse Fast Fourier Transform to reconstruct the probability density function $\rho(\Delta y)$ for Δy based on these values.*

The first auxiliary problem – of transforming ρ into μ, A_j, and B_j – is somewhat more complicated. Let us analyze how we can solve it.

Natural idea for the first auxiliary algorithm: use Least Squares. We want to approximate the given distribution $\rho(x)$ by a distribution $\rho_G(x)$ in the desired granular form. A reasonable idea is to use the Least Squares approximation, i.e., to find a distribution $\rho(x)$ for which the value

$$\int (\rho(x) - \rho_G(x))^2 \, dx$$

is the smallest possible.

Let us reformulate this idea in terms of the characteristic functions. The problem with the above

idea is that while for $\alpha = 1$ and $\alpha = 2$, we have explicit expressions for the corresponding probability density function $\rho_G(x)$, we do not have such an expression for any other α. Instead, we have an explicit expression for the characteristic function $\chi(\omega)$. It is therefore desirable to reformulate the above idea in terms of characteristic functions.

We want to approximate the characteristic function $\chi(\omega)$ by an expression $\chi_G(\omega)$ of the type $\exp\left(-\sum\limits_j c_j \cdot f_j(\omega)\right)$ for some fixed functions $f_j(\omega)$; in our case, these functions are $-\mathrm{i}\cdot\omega$, $|\omega|^{\alpha_j}$, and $\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}$.

This can be done, since, due to Parceval theorem, the least squares (L^2) difference

$$\int (\rho(x) - \rho_G(x))^2\,dx$$

between the corresponding pdfs $\rho(x)$ and $\rho_G(x)$ is proportional to the least squares difference between the characteristic functions:

$$\int (\rho(x) - \rho_G(x))^2\,dx =$$

$$\frac{1}{2\pi}\cdot\int (\chi(\omega) - \chi_G(\omega))^2\,d\omega.$$

So, minimizing the value

$$\int (\rho(x) - \rho_G(x))^2\,dx$$

is equivalent to minimizing the integral

$$I \stackrel{\text{def}}{=} \int (\chi(\omega) - \chi_G(\omega))^2\,d\omega.$$

How to approximate: computational challenge and its solution. The problem with the above formulation is that the Least Squares method is very efficient is we are looking for the coefficients of a linear dependence. However, in our case, the dependence of the expression $\chi_G(\omega)$ on the parameters μ, A_j, and B_j is non-linear, which makes computations complicated.

How can we simplify computations? We can borrow the idea from the case of normal distributions: in this case,

- we start with the maximum likelihood methods, in which we maximize the probability, and then
- we take negative logarithms of the pdfs – which results in the known Least Squares method (Sheskin 2011).

In our more general case too, if we take the negative logarithm of the characteristic function, we get a linear function of the unknowns:

$$-\ln(\chi_G(\omega)) =$$

$$-\mathrm{i}\cdot\mu_i\cdot\omega + \sum_{j=1}^{k} A_{ij}\cdot|\omega|^{\alpha_j} + \sum_{j=1}^{k} B_j\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}.$$

To use this idea, let us reformulate the objective function

$$\int (\chi(\omega) - \chi_G(\omega))^2\,d\omega$$

in terms of the difference between the negative logarithms. We are interested in situations in which the approximation is good, i.e., in which the difference $\varepsilon(\omega) \stackrel{\text{def}}{=} \chi_G(\omega) - \chi(\omega)$ is small. Then,

$$\chi_G(\omega) = \chi(\omega) + \varepsilon(\omega),$$

hence

$$-\ln(\chi_G(\omega)) = -\ln(\chi(\omega) + \varepsilon(\omega)) =$$

$$-\ln\left(\chi(\omega)\cdot\left(1 + \frac{\varepsilon(\omega)}{\chi(\omega)}\right)\right) =$$

$$-\ln(\chi(\omega)) - \ln\left(1 + \frac{\varepsilon(\omega)}{\chi(\omega)}\right).$$

Since $\varepsilon(\omega)$ is small, we can ignore terms which are quadratic and higher order in $\varepsilon(\omega)$ and get

$$\ln\left(1 + \frac{\varepsilon(\omega)}{\chi(\omega)}\right) \approx \frac{\varepsilon(\omega)}{\chi(\omega)}.$$

Thus, in this approximation,

$$(-\ln(\chi(\omega))) - (-\ln(\chi_G(\omega))) = \frac{\varepsilon(\omega)}{\chi(\omega)},$$

hence

$$\varepsilon(\omega) = \chi_G(\omega) - \chi(\omega) =$$

$$\chi(\omega)\cdot((-\ln(\chi_G(\omega))) - (-\ln(\chi(\omega)))),$$

so the minimized integral takes the form

$$I = \int (\chi(\omega) - \chi_G(\omega))^2\,d\omega =$$

$$\int \chi^2(\omega)\cdot((-\ln(\chi(\omega))) - (-\ln(\chi_G(\omega)))^2\,d\omega,$$

or, equivalently, the form

$$I = \int (f(\omega) - f_G(\omega))^2\,d\omega,$$

where we denoted

$$f(\omega) \stackrel{\text{def}}{=} -\chi(\omega)\cdot\ln(\chi(\omega))$$

and

$$f_G(\omega) \stackrel{\text{def}}{=} -\chi(\omega)\cdot\ln(\chi_G(\omega)).$$

In our case

$$f_G(\omega) = -\mathrm{i}\cdot\mu\cdot\omega\cdot\chi(\omega) + \sum_{j=1}^{k} A_j\cdot\chi(\omega)\cdot|\omega|^{\alpha_j} + \sum_{j=1}^{k} B_j\cdot\chi(\omega)\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}.$$

In other words, we need to find the coefficients μ, A_j, and B_j by applying the Least Squares method to the approximate equality

$$-\ln(\chi(\omega))\cdot\chi(\omega) \approx -\mathrm{i}\cdot\mu\cdot\omega\cdot\chi(\omega) + \sum_{j=1}^{k} A_j\cdot\chi(\omega)\cdot|\omega|^{\alpha_j} + \sum_{j=1}^{k} B_j\cdot\chi(\omega)\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}.$$

Thus, we arrive at the following algorithm.

Algorithm $A_{\rho\to G}$ for transforming ρ into μ, A_j, and B_j. *Let us assume that we are given:*

- *the values $\rho(t_0)$, $\rho(t_1)$, $\rho(t_2)$, ... of a pdf on a grid t_0, t_1, t_2, etc., and*
- *the selected values $\alpha_1,\dots,\alpha_k$.*

To find the values μ, $A_1,\dots,A_k$, and $B_1,\dots,B_k$ describing this probability distribution, we need to do the following:

- *first, we apply Fast Fourier transform to the values $\rho(t_i)$ and thus, get the values $\chi(\omega_k)$ of the corresponding characteristic function;*
- *then, to find the desired values μ, $A_1,\dots,A_k$, and $B_1,\dots,B_k$, we use the Least Squares method to solve the following system of linear equations:*

$$-\ln(\chi(\omega))\cdot\chi(\omega) \approx -\mathrm{i}\cdot\mu\cdot\omega\cdot\chi(\omega) + \sum_{j=1}^{k} A_j\cdot\chi(\omega)\cdot|\omega|^{\alpha_j} + \sum_{j=1}^{k} B_j\cdot\chi(\omega)\cdot\mathrm{sign}(\omega)\cdot|\omega|^{\alpha_j}.$$

Thus, we arrive at the following algorithm for solving our main problem in the case when the probability distributions are represented by their pdfs.

Granular-based algorithm $A^L_{\rho\,(\text{via } G)}$ for solving the pdf-based problem P^L_ρ. *Suppose that we are given:*

- *the pdfs $\rho_i(t_{ik})$ of n independent random variables*

$$\Delta x_1,\dots,\Delta x_n,$$

- *the coefficients $c_1,\dots,c_n$, and*
- *the parameters $\alpha_1,\dots,\alpha_k$.*

Then, to compute the pdf $\rho(\Delta y)$ of the expression

$$\Delta y = \sum_{i=1}^{n} c_i\cdot\Delta x_i,$$

we do the following:

- *first, to information about each variable Δx_i, we apply the algorithm $A_{\rho\to G}$ and get the corresponding values μ_i, $A_{i1},\dots,A_{ik}$, and $B_{i1},\dots,B_{ik}$;*
- *then, we use the algorithm A^L_G for solving the granular-based problem problem P^L_G and get the resulting values μ, $A_1,\dots,A_k$, and $B_1,\dots,B_k$;*
- *finally, we apply the algorithm $A_{G\to\rho}$ to compute the values $\rho(t_i)$ of the desired pdf $\rho(\Delta y)$.*

Comment. Both Fourier transform and Inverse Fourier Transform – which are parts of the algorithms $A_{\rho\to G}$ and $A_{G\to\rho}$ – require time $O(N\cdot\log(N))$. Thus, asymptotically, this algorithm requires the same time $O(N\cdot\log(N))$ as Algorithm A^S_χ based on the characteristic functions.

However, if instead of the pdf, we use the granular representation of the corresponding probability distributions, the corresponding algorithm P^L_G takes time $O(1)$ – much faster than the algorithm P^S_χ based on representing probability distributions by their characteristic functions.

5 Numerical Examples

We have tested our method on several examples, let us provide two such examples. In both examples, we used $k=3$, $\alpha_1=1$, $\alpha_2=1.5$, and $\alpha_3=2$.

To each of the two examples, we applied the following four algorithms for solving the main problem:

- the algorithms A^S_ρ and $A^S_{\rho\,(\text{via }\chi)}$ based on representing the probability distributions in terms of the probability density functions;
- the algorithm A^S_χ based on representing the probability distributions in terms of characteristic functions; and
- the algorithm A^L_G based on the granular representation of the probability distributions.

5.1 Example 1

Description of the example. In this example, we considered the following $n = 2$ case:

- the first measurement errors Δx_1 is normally distributed with 0 mean and standard deviation 1,
- the second measurement error Δx_2 has Laplace distribution, with probability density

$$\rho_2(\Delta x_2) = \frac{1}{2} \cdot \exp(-|\Delta x_2|),$$

- and $c_1 = c_2 = 1$.

The fact that $c_1 = c_2 = 1$ means that we want to find the probability distribution for

$$\Delta y = c_1 \cdot \Delta x_1 + c_2 \cdot \Delta x_2 = \Delta x_1 + \Delta x_2.$$

Applying the algorithm $A_\rho^{S(n=2)}$. In the pdf representation, we represented each of the two probability distributions by the values of the corresponding probability density function $\rho(x)$ on a grid of values from the interval $[-5, 5]$ of length 10. As a step h for this grid, we selected $h = 0.01$, so the overall number of grid points is $N = \dfrac{10}{0.01} = 1000$ (to be precise, there are 1001 grid points).

In this case, the algorithm $A_\rho^{S(n=2)}$ requires $N^2 = 1000^2 = 10^6$ (a million) computational steps.

Applying the algorithm $A_{\rho\ (\text{via }\chi)}^{S}$. First, we apply the Fast Fourier Transform to each of the pdfs to find the corresponding characteristic functions. The results are:

$$\chi_1(\omega) = \exp\left(-\frac{1}{2} \cdot \omega^2\right) \text{ and } \chi_2(\omega) = \frac{1}{1+\omega^2}.$$

This transformation requires $N \cdot \log_2(N)$ computational steps – i.e., in this case, when $\log_2(1000) \approx 10$, we need $10000 = 10^4$ computational steps.

Then, for each k, we multiplied $\chi_1(\omega_k)$ and $\chi_2(\omega_k)$ to get the corresponding value $\chi(\omega_k)$ of the characteristic function for Δy. This required $N = 1000$ computational steps.

Finally, we performed the inverse Fourier transform to compute the values of the corresponding pdf. This also required $N \cdot \log_2(N) \approx 10000$ computational steps. Thus, overall, we needed $10 + 1 + 10 = 21$ thousand computational steps, which is much fewer than the million steps needed for the straightforward algorithm $A_\rho^{S(n=2)}$.

Applying the algorithm A_χ^S. In the characteristic function representation, we represented each of the two probability distributions by the values of the corresponding characteristic functions

$$\chi_1(\omega_k) = \exp\left(-\frac{1}{2} \cdot \omega_k^2\right) \text{ and } \chi_2(\omega_k) = \frac{1}{1+\omega_k^2}.$$

To compute N values $\chi(\omega_k)$, we need to perform $N = 1000$ computational steps. This is much smaller than for both algorithms based on representing probability distributions in the pdf form.

Applying the algorithm A_G^S. In this case, we represented each of the two distributions in the granular form (we can get this form, e.g., by applying the auxiliary algorithm $A_{\rho\to G}$ to the pdfs):

$$\mu_1 = 0, \quad A_{11} = A_{12} = 0, \quad A_{13} = \frac{1}{2}, \quad B_{1j} = 0;$$

$$\mu_2 = 0, \quad A_{21} = -0.162, \quad A_{22} = 1.237,$$

$$A_{23} = -0.398, \quad B_{2j} = 0.$$

In accordance with the granular-based algorithm A_G^L, we computed the following values:

$$\mu = c_1 \cdot \mu_1 + c_2 \cdot \mu_2 = 0;$$

$$A_1 = |c_1|^{a_1} \cdot A_{11} + |c_2|^{a_1} \cdot A_{21} = -0.162,$$

$$A_2 = |c_1|^{a_2} \cdot A_{12} + |c_2|^{a_2} \cdot A_{22} = 1.237,$$

$$A_3 = |c_1|^{a_3} \cdot A_{13} + |c_2|^{a_3} \cdot A_{23} = 0.102,$$

$$B_1 = \text{sign}(c_1)|c_1|^{a_1} \cdot B_{11} + \text{sign}(c_2)|c_2|^{a_1} \cdot B_{21} = 0,$$

$$B_2 = \text{sign}(c_1)|c_1|^{a_2} \cdot B_{12} + \text{sign}(c_2)|c_2|^{a_2} \cdot B_{22} = 0,$$

$$B_1 = \text{sign}(c_1)|c_1|^{a_3} \cdot B_{13} + \text{sign}(c_2)|c_2|^{a_3} \cdot B_{23} = 0.$$

To compute all these values, we needed 3 raising-to-the-power, 12 multiplications, and 7 additions – the total of 22 computational steps, much fewer than when we represent probability distributions by their characteristic functions.

Comment. To compare the four computation results, we transformed them all into the pdf form and plotted them in Fig. 1. As one can see from this figure, the curves are practically indistinguishable. The mean square difference between every two curves is smaller than 0.01 – in good accordance with the fact that we used the grid with step $h = 0.01$ to approximate the corresponding functions.

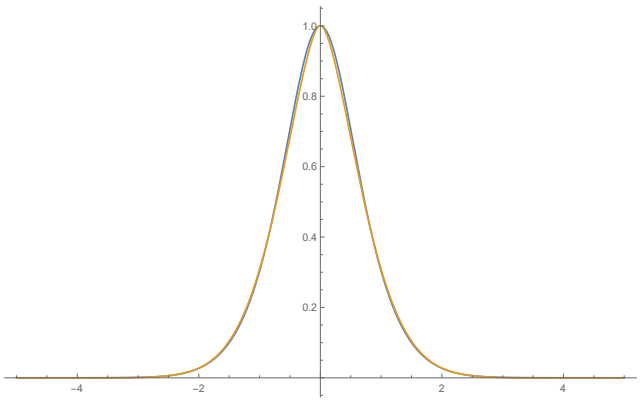

Fig. 1 Example 1: the resulting pdf

5.2 Example 2

Description of the example. In this example, we also took $n = 2$. Here:

- both Δx_1 and Δx_2 have Laplace distributions, with $\rho_1(\Delta x_2) = \exp(-2|\Delta x_2|)$ and

$$\rho_2(\Delta x_2) = \frac{3}{2} \cdot \exp(-3|\Delta x_2|),$$

- and $c_1 = c_2 = 1$.

We want to find the probability distribution for

$$\Delta y = c_1 \cdot \Delta x_1 + c_2 \cdot \Delta x_2 = \Delta x_1 + \Delta x_2.$$

Applying the algorithm $A_\rho^{S(n=2)}$. In the pdf representation, we represented each of the two probability distributions by the values of the corresponding probability density function $\rho(x)$ on a grid of values from the interval $[-5, 5]$ of length 10. As a step h for this grid, we selected $h = 0.01$, so the overall number of grid points is $N = \dfrac{10}{0.01} = 1000$ (to be precise, there are 1001 grid points).

In this case, the algorithm $A_\rho^{S(n=2)}$ requires $N^2 = 1000^2 = 10^6$ (a million) computational steps.

Applying the algorithm $A_{\rho\ (\text{via }\chi)}^{S}$. First, we apply the Fast Fourier Transform to each of the pdfs to find the corresponding characteristic functions. The results are:

$$\chi_1(\omega) = \frac{4}{4+\omega^2} \text{ and } \chi_2(\omega) = \frac{9}{9+\omega^2}.$$

This transformation requires $N \cdot \log_2(N)$ computational steps – i.e., in this case, when $\log_2(1000)$ is approximately 10, this means $10000 = 10^4$ computational steps.

Then, for each k, we multiplied $\chi_1(\omega_k)$ and $\chi_2(\omega_k)$ to get the corresponding value $\chi(\omega_k)$ of the characteristic function for Δy. This required $N = 1000$ computational steps.

Finally, we performed the inverse Fourier transform to compute the values of the corresponding pdf. This also required $N \cdot \log_2(N) \approx 10000$ computational steps. Thus, overall, we needed $10 + 1 + 10 = 21$ thousand computational steps, which is much fewer than the million steps needed for the straightforward algorithm $A_\rho^{S(n=2)}$.

Applying the algorithm A_χ^S. In the characteristic function representation, we represented each of the two probability distributions by the values of the corresponding characteristic functions

$$\chi_1(\omega_k) = \frac{4}{4+\omega_k^2} \text{ and } \chi_2(\omega_k) = \frac{9}{9+\omega_k^2}.$$

To compute N values $\chi(\omega_k)$, we need to perform $N = 1000$ computational steps. This is much smaller than for both algorithms based on representing probability distributions in the pdf form.

Applying the algorithm A_G^L. In this case, we represented each of the two distributions in the granular form (we can get this form, e.g., by applying the auxiliary algorithm $A_{\rho\to G}$ to the pdfs):

$$\mu_1 = 0, \quad A_{11} = 0.283, \quad A_{12} = -0.685,$$
$$A_{13} = 0.171, \quad B_{1j} = 0;$$
$$\mu_2 = 0, \quad A_{21} = 0.156, \quad A_{22} = -0.327,$$
$$A_{23} = 0.061, \quad B_{2j} = 0.$$

In accordance with the granular-based algorithm A_G^L, we computed the following values:

$$\mu = c_1 \cdot \mu_1 + c_2 \cdot \mu_2 = 0;$$
$$A_1 = |c_1|^{a_1} \cdot A_{11} + |c_2|^{a_1} \cdot A_{21} = 0.439,$$
$$A_2 = |c_1|^{a_2} \cdot A_{12} + |c_2|^{a_2} \cdot A_{22} = -1.013,$$
$$A_3 = |c_1|^{a_3} \cdot A_{13} + |c_2|^{a_3} \cdot A_{23} = 0.232,$$
$$B_1 = \text{sign}(c_1)|c_1|^{a_1} \cdot B_{11} + \text{sign}(c_2)|c_2|^{a_1} \cdot B_{21} = 0,$$
$$B_2 = \text{sign}(c_1)|c_1|^{a_2} \cdot B_{12} + \text{sign}(c_2)|c_2|^{a_2} \cdot B_{22} = 0,$$
$$B_1 = \text{sign}(c_1)|c_1|^{a_3} \cdot B_{13} + \text{sign}(c_2)|c_2|^{a_3} \cdot B_{23} = 0.$$

To compute all these values, we needed 3 raising-to-the-power, 12 multiplications, and 7 additions – the total of 22 computational steps, much fewer than when we represent probability distributions by their characteristic functions.

Comment. To compare the four computation results, we transformed them all into the pdf form. The mean square difference between every two curves is approximately equal to 0.008 – also in good accordance with the fact that we used the grid with step $h = 0.01$ to approximate the corresponding functions.

6 Conclusions

In general, data processing means that:

- we know the values of the inputs $x_1, \ldots, x_n$, and
- we apply an algorithm $f(x_1, \ldots, x_n)$ to these inputs to get the desired value y.

Often, the inputs come from measurements, and measurements are never absolutely accurate: with certain probabilities, we can have different values of the difference $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ between the measurement result $\widetilde{x}_i$ and the actual (unknown) value of the corresponding quantity. Thus, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of applying the data processing algorithm to the measurement results is, in general, different from the desired value

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

So, when processing data under such probabilistic uncertainty, it is desirable to get not only the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, but also the probability distribution of the accuracy $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ of this data processing result. There exist several techniques for solving this problem; however, often, these techniques take too much computation time. In such situations, it is desirable to come up with a faster algorithm for data processing under probabilistic uncertainty.

In this paper, we show that it is possible to design such an algorithm if we apply granularity ideas. Specifically, we propose to do the following:

- decompose each given distribution into granules of appropriate types,
- process the resulting granular data type-by-type, and then
- combine the results of type-by-type data processing into the desired probability distribution.

As a result, we indeed get a faster algorithm for data processing under probabilistic uncertainty.

Acknowledgements The authors are thankful to Witold Pedrycz and Shyi-Ming Chen for their encouragement, and to the anonymous referees for valuable suggestions.

Conflict of Interest Section

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to Algorithms. MIT Press, Cambridge, Massachusetts

Kovalerchuk B, Kreinovich V (2017) Concepts of solutions of uncertain equations with intervals, probabilities and fuzzy sets for applied tasks. Granular Computing 2(3):121–130

Kreinovich V (2016) Solving equations (and systems of equations) under uncertainty: how different practical problems lead to different mathematical and computational formulations. Granular Computing 1(3):171–179

Pedrycz W, Chen SM (2011) Granular Computing and Intelligent Systems: Design with Information Granules of High Order and High Type. Springer, Heidelberg, Germany

Pedrycz W, Chen SM (2015) Information Granularity, Big Data, and Computational Intelligence. Springer, Heidelberg, Germany

Pedrycz W, Chen SM (2015) Granular Computing and Decision-Making: Interactive and Iterative Approaches. Springer, Heidelberg, Germany

Piegat A, Landowski M (2018) Solving different practical granular problems under the same system of equations. Granular Computing 3(1):39–48

Rabinovich SG (2005) Measurement Errors and Uncertainty: Theory and Practice. Springer Verlag, Berlin

Sheskin DJ (2011) Handbook of Parametric and Nonparametric Statistical Procedures. Chapman and Hall/CRC, Boca Raton, Florida

Stylios CD, Pownuk A, Kreinovich V (2015) Sometimes, it is beneficial to process different types of uncertainty separately, In: Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2015 and 5th World Conference on Soft Computing, Redmond, Washington, August 17–19, 2015

A Appendix: Non-Uniform Distribution of α_j is Better

Idea. If we select two values α_j too close to each other, there will be too much correlation between them, so adding the function corresponding to the second value does not add much information to what we know from a function corresponding to the first value.

We are approximating a general function (logarithm of a characteristic function) as a linear combination of functions $|t|^{\alpha_j}$. If two values α_j and α_{j+1} are close, then the function $|t|^{\alpha_{j+1}}$ can be well approximated by a term linear in $|t|^{\alpha_j}$, thus, the term proportional to $|t|^{\alpha_{j+1}}$ is not needed.

It therefore makes sense to select the values α_j in such as way that for each j, the part of $|t|^{\alpha_{j+1}}$ that cannot be approximated by terms proportional to $|t|^{\alpha_j}$ should be the largest possible.

Let us reformulate this idea in precise terms. For every two functions $f(t)$ and $g(t)$, the part of $g(t)$ which cannot be represented by terms $a \cdot f(t)$ (proportional to $f(t)$) can be described as follows. It is reasonable to describe the difference between the two functions $f(t)$ and $g(t)$ by the least squares (L^2) metric $\int (f(t) - g(t))^2\, dt$. In these terms, the value of a function itself itself can be described as its distance from 0, i.e., as $\int (f(t))^2\, dt$.

When we approximate a function $g(t)$ by a term $a \cdot f(t)$, then the remainder $g(t) - a \cdot f(t)$ has the value

$$\int (g(t) - a \cdot f(t))^2\, dt.$$

The best approximation occurs when this value is the smallest, i.e., when it is equal to $\min_a \int (g(t) - a \cdot f(t))^2\, dt$. Out of

the original value $\int (g(t))^2\,dt$, we have unrepresented the part equal to $\min\limits_a \int (g(t) - a\cdot f(t))^2\,dt$. Thus, the relative size of what cannot be represented by terms $a\cdot f(t)$ can be defined as a ratio

$$R(f(t), g(t)) = \frac{\min\limits_a \int (g(t) - a\cdot f(t))^2\,dt}{\int (g(t))^2\,dt}.$$

Let us simplify the resulting expression. This expression can be simplified if we find the explicit expression for a for which the value $\int (g(t) - a\cdot f(t))^2\,dt$ is the smallest possible. Differentiating the minimized expression with respect to a and equating the derivative to 0, we conclude that

$$-\int (g(t) - a\cdot f(t))\cdot f(t)\,dt = 0,$$

i.e., that

$$a\cdot \int (f(t))^2\,dt = \int f(t)\cdot g(t)\,dt,$$

and

$$a = \frac{\int f(t)\cdot g(t)\,dt}{\int (f(t))^2\,dt}.$$

For this a, the value $\int (g(t) - a\cdot f(t))^2\,dt$ takes the form

$$\int (g(t) - a\cdot f(t))^2\,dt =$$

$$\int (g(t))^2\,dt - 2a\cdot \int f(t)\cdot g(t)\,dt + a^2\cdot \int (f(t))\,dt.$$

Substituting the above expression for a into this formula, we conclude that

$$\int (g(t) - a\cdot f(t))^2\,dt =$$

$$\int (g(t))^2\,dt - \frac{2(\int f(t)\cdot g(t)\,dt)^2}{\int (f(t))^2\,dt} + \frac{(\int f(t)\cdot g(t)\,dt)^2}{\int (f(t))^2\,dt},$$

i.e., that

$$\int (g(t) - a\cdot f(t))^2\,dt = \int (g(t))^2\,dt - \frac{(\int f(t)\cdot g(t)\,dt)^2}{\int (f(t))^2\,dt}.$$

Thus, the desired ratio takes the form

$$R(f(t), g(t)) \stackrel{\text{def}}{=} \frac{\min\limits_a \int (g(t) - a\cdot f(t))^2\,dt}{\int (g(t))^2\,dt} =$$

$$1 - \frac{(\int f(t)\cdot g(t)\,dt)^2}{(\int (f(t))^2\,dt)\cdot(\int (g(t))^2\,dt)}.$$

Thus, we arrive at the following optimization problem.

Resulting optimization problem. To make sure that the above remainders are as large as possible, it makes sense to find the values $\alpha_1^{\text{opt}} < \ldots < \alpha_k^{\text{opt}}$ that maximize the smallest of the remainders between the functions $f(t) = |t|^{\alpha_j}$ and $g(t) = |t|^{\alpha_{j+1}}$:

$$\min_j R\left(|t|^{\alpha_j^{\text{opt}}}, |t|^{\alpha_{j+1}^{\text{opt}}}\right) = \max_{\alpha_1<\ldots<\alpha_k} \min_j R(|t|^{\alpha_j}, |t|^{\alpha_{j+1}}).$$

Solving the optimization problem. Let us consider an interval $[-T, T]$ for some T. Since the function is symmetric, it is sufficient to consider the values from $[0, T]$.

For $f(t) = t^\alpha$ and $g(t) = t^\beta$, the integral in the numerator of the ratio is equal to

$$\int_0^T f(t)\cdot g(t)\,dt = \int_0^T t^\alpha\cdot t^\beta\,dt = \int_0^T t^{\alpha+\beta}\,dt = \frac{T^{\alpha+\beta+1}}{\alpha+\beta+1}.$$

Similarly, the integrals in the denominator take the form

$$\int_0^T f^2(t)\,dt = \int_0^T t^{2\alpha}\,dt = \frac{T^{2\alpha+1}}{2\alpha+1}$$

and

$$\int_0^T g^2(t)\,dt = \int_0^T t^{2\beta}\,dt = \frac{T^{2\beta+1}}{2\beta+1},$$

so

$$R = 1 - \frac{\dfrac{T^{2(\alpha+\beta+1)}}{(\alpha+\beta+1)^2}}{\dfrac{T^{2\alpha+1}}{2\alpha+1}\cdot\dfrac{T^{2\beta+1}}{2\beta+1}}.$$

One can see that the powers of T cancel each other, and we get

$$R = 1 - \frac{(2\alpha+1)\cdot(2\beta+1)}{(\alpha+\beta+1)^2},$$

or, equivalently, if we denote $r \stackrel{\text{def}}{=} \dfrac{\beta+0.5}{\alpha+0.5}$, we get

$$R = R(r) \stackrel{\text{def}}{=} 1 - 4\cdot\frac{r}{(1+r)^2}.$$

The derivative of the function $R(r)$ is equal to

$$\frac{dR}{dr} = -4\cdot\frac{(1+r)^2 - 2\cdot(1+r)}{(1+r)^4} = -4\cdot\frac{(1+r)\cdot(1+r-2)}{(1+r)^4} =$$

$$4\cdot\frac{(1+r)\cdot(r-1)}{(1+r)^4} = 4\cdot\frac{r-1}{(1+r)^3}.$$

So this derivative is positive for all $r > 1$. Thus, the function $R(r)$ is monotonically increasing, and looking for the values α_j^{opt} for which $\min\limits_j R(|t|^{\alpha_j}, |t|^{\alpha_{j+1}})$ is the largest is equivalent to looking for the values α_j^{opt} for which the smallest $\min\limits_j \dfrac{\alpha_{j+1}+0.5}{\alpha_j+0.5}$ of the ratios $r = \dfrac{\alpha_{j+1}+0.5}{\alpha_j+0.5}$ attains the largest possible value:

$$\min_j \frac{\alpha_{j+1}^{\text{opt}}+0.5}{\alpha_j^{\text{opt}}+0.5} = \max_{\alpha_1<\ldots<\alpha_k} \min_j \frac{\alpha_{j+1}+0.5}{\alpha_j+0.5}.$$

One can check that this happens when

$$\alpha_j + 0.5 = 1.5\cdot\left(\frac{5}{3}\right)^{(j-1)/(k-1)}.$$

Indeed, in this case, $\min\limits_j \dfrac{\alpha_{j+1}+0.5}{\alpha_j+0.5} = \left(\dfrac{5}{3}\right)^{1/(k-1)}$. We cannot have it larger: if we had $\min\limits_j \dfrac{\alpha_{j+1}+0.5}{\alpha_j+0.5} > \left(\dfrac{5}{3}\right)^{k-1}$, then we would have $\dfrac{\alpha_{j+1}+0.5}{\alpha_j+0.5} > \left(\dfrac{5}{3}\right)^{k-1}$ for all j. Here,

$$\alpha_k + 0.5 = (\alpha_1+0.5)\cdot\frac{\alpha_2+0.5}{\alpha_1+0.5}\cdot\frac{\alpha_3+0.5}{\alpha_2+0.5}\cdot\ldots\cdot\frac{\alpha_k+0.5}{\alpha_{k-1}+0.5}.$$

The first factor $\alpha_1 + 0.5$ is ≥ 1.5, each of the other $k-1$ terms is greater than $\left(\frac{5}{3}\right)^{1/(k-1)}$, so for their product, we get

$$\alpha_k + 0.5 > 1.5 \cdot \left(\left(\frac{5}{3}\right)^{1/(k-1)}\right)^{k-1} = 1.5 \cdot \frac{5}{3} = 2.5,$$

while we assumed that all the values α_j are from the interval $[1, 2]$, and so, we should have $\alpha_k + 0.5 \leq 2.5$.

Resulting optimal values of α_j. Thus, the optimal way is to not to take the values uniformly distributed on the interval $[1, 2]$, but rather take the values

$$\alpha_j^{\text{opt}} = 1.5 \cdot \left(\frac{5}{3}\right)^{(j-1)/(k-1)} - 0.5$$

for which the logarithms $\ln(\alpha_j^{\text{opt}} + 0.5) = \frac{j-1}{k-1} \cdot \ln\left(\frac{5}{3}\right)$ are uniformly distributed.

Comment. It is worth mentioning that there is intriguing connection between these values α_j and music: for example, the twelve notes on a usual Western octave correspond to the following frequencies:

- the first note corresponds to the frequency f_1,
- the second note corresponds to the frequency

$$f_2 = f_1 \cdot 2^{1/12},$$

- the third note correspond to the frequency

$$f_3 = f_1 \cdot 2^{2/12},$$

- $\ldots$,
- the last note corresponds to the frequency

$$f_{12} = f_1 \cdot 2^{11/12},$$

  and
- the first note of the next octave corresponds to the frequency $f_{13} = f_1 \cdot 2$.

For these frequencies, the logarithms $\ln(f_j)$ are uniformly distributed.

Similar formulas exist for five-note and other octaves typical for some Oriental musical traditions.