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 In many real-life situations, we need to process measurement results. Due to inevitable measurement errors, the measurement results are, in general, somewhat different from the actual (unknown) values of the corresponding quantities. As a result, the value that we obtained by processing the measurement results is, in general, different from what we would have got if we were able to process the actual (exact) values. In many practical situations, it is important to know how accurate is the resulting estimate. In such situations, processing data under probabilistic uncertainty involves not only processing the measurement results, but also providing a probability distribution describing how accurate is the result of this processing.

There exist several algorithms for such data processing under probabilistic uncertainty, but the existing algorithms often require too much computation time. To speed up the corresponding computations, 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.

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

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

1 Introduction

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.

Therefore, 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., [5].

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.

Because of this practical importance, there are many techniques for data processing under uncertainty, both in the case when the relation is given in the form of an explicit algorithm (see, e.g., [5] and references therein), and in the case when this relation is given in terms of systems of equations; see, e.g., [2–4].

Measurement errors are usually relatively small. Measurement errors are usually assumed to be relatively small, so that terms quadratic in measurement errors can be safely ignored [5].

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

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

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

In general, the simplified expression (1) enables us to estimate the uncertainty in the result y of data processing based on the known information about the uncertainties Δx_i.

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

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., [6]), 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 distribution is often different from Gaussian.

Measurement errors corresponding to different variables are usually independent.

We thus arrive at the following problem.

Data processing under probabilistic uncertainty: formulation of 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 distribution of each of the variables Δx_i.

Based on this information, we can compute the values $c_i = \dfrac{\partial f}{\partial x_i}_{|x_j = \widetilde{x}_j}$.

We need to find the probability distribution of the quantity

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

How to simplify the corresponding computational problem. Usually, when we know the probability distribution of a quantity Δx_i, then, for each real value c_i, it is easy to find the probability distribution for the quantity $t_i \stackrel{\text{def}}{=} c_i \cdot \Delta x_i$. In terms of these products t_i, the desired measurement error has a simpler form $\Delta y = \sum\limits_{i=1}^{n} t_i$. Thus, if we take this easiness into account, we arrive at the following problem:

– we know the probability distributions of each of n independent random variables $t_1, \ldots, t_n$, and
– we are interested in the probability distribution of their sum $t = \sum\limits_{i=1}^{n} t_i$.

The simplest case of this problem is when $n = 2$. In this case:

– we know the probability distributions of each of two independent random variables t_1 and t_2, and
– we are interested in 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 easily solve the original more general problem as well:

– 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.

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

So, to solve our original problem of data processing under probabilistic uncertainty, it is sufficient to solve the following computational problem:

Basic computational problem.

– We know the probability distributions of each of two independent random variables t_1 and t_2.
– We are interested in the probability distribution of their sum $t = t_1 + t_2$.

Straightforward (naive) approach to solving the basic computational problem. The usual way to represent a probability distribution is by its probability density function (pdf) $\rho(z)$. It is therefore reasonable to represent this information in the computer by values $\rho(z_i)$ 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.

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 [6]

$$\rho(t) = \int \rho_1(t_1) \cdot \rho_2(t - t_1)\, dt_1.$$

If we know:

– the values $\rho_1(t_{1i})$ corresponding to $t_{1i} = t_{10} + i \cdot h$ and
– the values $\rho_2(t_{2i})$ corresponding to $t_{2i} = t_{20} + i \cdot h$,

then, to find the values of the desired function $\rho(t)$, we can approximate the above integral by the integral sum. Specifically, or each value $v_k = v_0 + k \cdot h$, where $v_0 = t_{10} + t_{20}$, we have

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

Thus, we arrive at the following straightforward algorithm:

– we start with the values $\rho_i(t_{ij})$ describing the original probability distributions, and
– we compute the desired values $\rho(v_k)$ by using the formula (2).

How much computation time do we need to perform all these computations? 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 $N = (\overline{t} - \underline{t})/h$.

For large N, performing N^2 steps is feasible but rather slow; see, e.g., [1]. It is therefore desirable to be able to solve the basic computational problem faster.

What is known: a faster algorithm for solving the basic computational problem – and thus, for data processing under probabilistic uncertainty. In practice, users apply a faster algorithm which 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 denotes the expected value.

The possibility of faster computation comes from the fact that, 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, that we have

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

i.e., $\chi(\omega) = \chi_1(\omega)\cdot\chi_2(\omega)$, where we denoted $\chi(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i}\cdot\omega\cdot t)]$ and $\chi_i(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i}\cdot\omega\cdot t_i)]$.

So, if we represent each characteristic function by its values of a grid $\omega_k = \omega_0 + k\cdot h$, we conclude that

$$\chi(\omega_k) = \chi_1(\omega_k)\cdot\chi_2(\omega_k). \tag{3}$$

Hence, we arrive at the following algorithm:

– we start with $2N$ values $\chi_i(\omega_k)$ describing the probability distributions for t_1 and t_2;
– then, we apply the formula (3) to compute the values $\chi(\omega_k)$.

This algorithm requires one multiplication for each of N values $\chi(\omega_k)$. So, overall, this algorithm requires N computational steps – which is, for large N, much faster than the N^2 steps needed for the straightforward algorithm.

Comment. Of course:

– if the original information about the probability distributions comes in terms of the corresponding probability density functions,
– then we still need to compute the corresponding characteristic functions.

And:

– if we want the probability density function for t,
– then we have to recover it from the t's characteristic function.

For both transformation – from the probability density function to characteristic function and back – we can use the Fast Fourier Transform algorithm (see, e.g., [1]) that takes only $N\cdot\ln(N)$ steps.

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

Remaining problem: can we do it even faster? For large N, the time N needed for point-wise multiplication is still rather huge.

It is therefore reasonable to look for the ways to make these computations faster. This is what we do in this paper.

2 Main Idea: Using Granularity

Processing can be faster if both distributions are normal. If both t_i are normally distributed, then we do not need to perform N computational steps. Indeed, we know that the sum of two normal distributions with mean μ_i and variances V_i is also normal, with mean $\mu = \mu_1 + \mu_2$ and variance $V = V_1 + V_2$.

In this case, if we describe each distribution t_i by its mean and its variance, then to find a similar representation for the sum $t = t_1 + t_2$, we need only two computational steps – instead of N steps needed in the general case.

Other cases when we can speed up data processing. Same holds for any *infinitely divisible* distribution, with characteristic function

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

For example:

– For $\alpha = 2$ and $B = 0$, we get normal distribution.
– For $\alpha = 1$ and $B = 0$, we get Cauchy distribution, 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 $\Delta > 0$.

Indeed, in this case, once we know the distributions for t_1 and t_2, then, based on the corresponding characteristic functions

$$\chi_1(\omega) = \exp(\mathrm{i} \cdot \mu_1 \cdot \omega - A_1 \cdot |\omega|^\alpha - B_1 \cdot \operatorname{sign}(\omega) \cdot |\omega|^\alpha)$$

and

$$\chi_2(\omega) = \exp(\mathrm{i} \cdot \mu_2 \cdot \omega - A_2 \cdot |\omega|^\alpha - B_2 \cdot \operatorname{sign}(\omega) \cdot |\omega|^\alpha),$$

we can conclude that the characteristic function $\chi(\omega)$ for the sum $t_1 + t_2$ has the form

$$\chi(\omega) = \chi_1(\omega) \cdot \chi_2(\omega) = \exp(\mathrm{i} \cdot \mu \cdot \omega - A \cdot |\omega|^\alpha - B \cdot \operatorname{sign}(\omega) \cdot |\omega|^\alpha),$$

where $\mu = \mu_1 + \mu_2$, $A = A_1 + A_2$, and $B = B_1 + B_2$.

So, in this case too:

– if we describe each probability distribution t_i by the two corresponding parameters μ_i, A_i, and B_i,
– then, to compute the corresponding parameters μ, A, and B for the sum $t = t_1 + t_2$, we need three computational steps instead of N:
  – one step to add the means μ_i, and
  – two more steps to add the values A_i and B_i.

Granularity-based idea. Our idea is to artificially decompose the original probability distributions into several appropriate granules, i.e., specifically:

– to select several values $\alpha_1, \ldots, \alpha_k$ – e.g., $\alpha_1 = 1$ and $\alpha_2 = 2$, and
– to approximate each random variable t_i by a sum

$$t_{a,i} = r_{i1} + \ldots + t_{ij} + \ldots + r_{ik}$$

of infinitely divisible random variables r_{ij} corresponding to the selected values of α_j.

The characteristic function $\chi_{ij}(\omega)$ for 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 $\chi_{a,i}(\omega)$ of the sum $t_{a,i} = \sum\limits_{j=1}^{k} r_{ik}$ is equal to the product

$$\chi_{a,i}(\omega) = \prod_{j=1}^{k} \chi_{ij}(\omega) =$$

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

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

From $\chi_1(\omega) \approx \chi_{a,1}(\omega)$ and $\chi_2(\omega) \approx \chi_{a,2}(\omega)$, we conclude that the characteristic function $\chi(\omega) = \chi_1(\omega) \cdot \chi_2(\omega)$ for the sum $t = t_1 + t_2$ is approximately equal to the product of the approximating characteristic functions:

$$\chi(\omega) \approx \chi_a(\omega) \stackrel{\text{def}}{=} \chi_{a,2}(\omega) \cdot \chi_{a,2}(\omega) =$$

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

where $\mu = \mu_1 + \mu_2$, $A_j = A_{1j} + A_{2j}$, and $B_j = B_{j1} + B_{j2}$.

In this case, if we represent each of the random variables t_i by $2k+1$ values μ_i, $A_{i1}, \ldots, A_{ik}$, $B_{i1}, \ldots, B_{ik}$, hen to find the a similar representation for the sum t, we need to perform $2k+1$ arithmetic operations instead of N:

– one addition to compute μ

– and $2k$ additions to compute $2k$ values $A_1, \ldots, A_k, B_1, \ldots, B_k$.

In other words, we arrive at the following algorithm:

Resulting algorithm for solving the basic computational problem. We are given:

– the values μ_1, A_{11}, $\ldots$, A_{1k}, B_{11}, $\ldots$, B_{1k} that represent the random variable t_1, and
– the values μ_2, A_{21}, $\ldots$, A_{2k}, B_{21}, $\ldots$, B_{2k} that represent the random variable t_2.

To find the values μ, A_1, $\ldots$, A_k, B_1, $\ldots$, B_k corresponding to the sum $t = t_1 + t_2$, we perform the following operation:

– first, we compute $\mu = \mu_1 + \mu_2$;
– then, for each value j from 1 to k, we compute $A_j = A_{j1} + A_{j2}$ and $B_j = B_{j1} + B_{j2}$.

This algorithm requires $2k+1$ computational steps. So, for large N, this algorithm is much faster than the currently used algorithm based on characteristic functions.

From the basic computational problem to the original problem of data processing under probabilistic uncertainty. If we know the characteristic function $\chi'_i(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot \Delta x_i)]$ for Δx_i, then the characteristic function for $t_i = c_i \cdot \Delta x_i$ is equal to

$$\chi_i(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot c_i \cdot \Delta x_i)] = E[\mathrm{i} \cdot (\omega \cdot c_i) \cdot \Delta x_i] = \chi'_i(c_i \cdot \omega).$$

Thus, if

$$\chi'_i(\omega) \approx \exp\left(\mathrm{i} \cdot \mu'_i \cdot \omega - \sum_{j=1}^{k} A'_{ij} \cdot |\omega|^{\alpha_j} - \sum_{j=1}^{k} B_j \cdot \mathrm{sign}(\omega) \cdot |\omega|^{\alpha_j}\right),$$

we get

$$\chi_i(\omega) = \chi'_i(c_i \cdot \omega) \approx$$
$$\exp\left(\mathrm{i} \cdot c_i \cdot \mu'_i \cdot \omega - \sum_{j=1}^{k} A'_{ij} \cdot |c_i|^{\alpha_j} \cdot |\omega|^{\alpha_j} - \right.$$
$$\left. \sum_{j=1}^{k} B_k \cdot \mathrm{sign}(c_i) \cdot |c_i|^{\alpha_j} \cdot \mathrm{sign}(\omega) \cdot |\omega|^{\alpha_j}\right).$$

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

Since for the sum of independent random variables, the corresponding coefficients simply add, we arrive at the following algorithm.

Resulting algorithm for solving the original problem – of data processing under probabilistic uncertainty. 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, $A_1, \ldots, A_k$, $B_1, \ldots, B_k$ corresponding to $\Delta y = \sum\limits_{i=1}^{n} c_i \cdot \Delta x_i$, we perform the following operations:

- first, we compute $\mu = \sum\limits_{i=1}^{n} c_i \cdot \mu'_i$;
- then, for each value j from 1 to k, we compute

$$A_j = \sum_{i=1}^{n} |c_i|^{\alpha_j} \cdot A'_{ji} \text{ and } B_j = \sum_{i=1}^{n} \operatorname{sign}(c_i) \cdot |c_i|^{\alpha_j} \cdot B'_{ij}.$$

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

Comment. It is worth mentioning that a similar idea can be applied to the case of fuzzy uncertainty; see [7] for details.

How to go from probability density function to this representation and back. Natural questions are:

- What if the probability distributions for t_i are given not in the above form, but in the more traditional form of probability density functions?
- What if what we want is not the parameters μ, A_j, and B_j, but rather a probability density function for the sum t?

Thus, we have two auxiliary problems:

- first, we need to be able to approximate a given distribution $\rho_i(\Delta x_i)$ for Δ_i by the above expression – i.e., to find the corresponding values μ'_i, A'_{ij}, and B'_{ij};
- second, we need to be able, given the values μ, A_j, and B_j, to find the probability density function corresponding to the sum t.

The second auxiliary problem is reasonably straightforward:

- we form a characteristic function for Δy, i.e., we compute the values

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

  for values $\omega_k = \omega_0 + k \cdot h$ on a grid, and 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 is somewhat more complicated. Let us analyze how we can solve it.

3 Analysis of the First Auxiliary Problem

Natural idea: use Least Squares. We want to approximate the actual distribution $\rho_i(\Delta x_i)$ for each of the variables Δx_i by an approximate approximate distribution $\rho_{a,i}(\Delta x_i)$. A reasonable idea is to use the Least Squares approximation, i.e., to find a distribution $\rho_{a,i}(\Delta x_i)$ for which the value $\int(\rho_i(\Delta x_i) - \rho_{a,i}(\Delta x_i))^2\, d(\Delta x_i)$ 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 $\rho_a(t)$, 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 $\chi_i(\omega)$ by an expression $\chi_{a,i}(\omega)$ 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 $\int(\rho_i(\Delta x_i) - \rho_{a,i}(\Delta x_i))^2\, d(\Delta x_i)$ between the corresponding pdfs $\rho_i(\Delta x_i)$ and $\rho_{a,i}(\Delta x_i)$ is proportional to the least squares difference between the characteristic functions:

$$\int(\rho_i(\Delta x_i) - \rho_{a,i}(\Delta x_i))^2\, d(\Delta x_i) = \frac{1}{2\pi}\cdot\int(\chi_i(\omega) - \chi_{a,i}(\omega))^2\, d\omega.$$

So, minimizing the value $\int(\rho_i(\Delta x_i) - \rho_{a,i}(\Delta x_i))^2\, d(\Delta x_i)$ is equivalent to minimizing the integral

$$I \stackrel{\text{def}}{=} \int(\chi_i(\omega) - \chi_{a,i}(\omega))^2\, d\omega.$$

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 $\chi_{a,i}(\omega)$ on the parameters μ_i and A_{ij} 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 [6].

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

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

To use this idea, let us reformulate the objective function

$$\int (\chi_i(\omega) - \chi_{a,i}(\omega))^2\, d\omega$$

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 $\varepsilon_i(\omega) \stackrel{\text{def}}{=} \chi_{a,i}(\omega) - \chi_i(\omega)$ is small. Then, $\chi_{a,i}(\omega) = \chi_i(\omega) + \varepsilon_i(\omega)$, hence

$$-\ln(\chi_{a,i}(\omega)) = -\ln(\chi_i(\omega) + \varepsilon_i(\omega)) = -\ln\left(\chi_i(\omega) \cdot \left(1 + \frac{\varepsilon_i(\omega)}{\chi_i(\omega)}\right)\right) =$$

$$-\ln(\chi_i(\omega)) - \ln\left(1 + \frac{\varepsilon_i(\omega)}{\chi_i(\omega)}\right).$$

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

$$\ln\left(1 + \frac{\varepsilon_i(\omega)}{\chi_i(\omega)}\right) \approx \frac{\varepsilon_i(\omega)}{\chi_i(\omega)}.$$

Thus, in this approximation,

$$(-\ln(\chi_i(\omega))) - (-\ln(\chi_{a,i}(\omega))) = \frac{\varepsilon_i(\omega)}{\chi_i(\omega)},$$

hence

$$\varepsilon_i(\omega) = \chi_{a,i}(\omega) - \chi_i(\omega) = \chi_i(\omega) \cdot ((-\ln(\chi_{a,i}(\omega))) - (-\ln(\chi_i(\omega))),$$

so the minimized integral takes the form

$$I = \int (\chi_i(\omega) - \chi_{a,i}(\omega))^2\, d\omega = \int \chi_i^2(\omega) \cdot ((-\ln(\chi_i(\omega))) - (-\ln(\chi_{a,i}(\omega)))^2\, d\omega,$$

or, equivalently, the form

$$I = \int (f_i(\omega) - f_{a,i}(\omega))^2\, d\omega,$$

where we denoted

$$f_i(\omega) \stackrel{\text{def}}{=} -\chi_i(\omega) \cdot \ln(\chi_i(\omega))$$

and

$$f_{a,i}(\omega) \stackrel{\text{def}}{=} -\chi_i(\omega) \cdot \ln(\chi_{a,i}(\omega)).$$

In our case

$$f_{a,i}(\omega) = -\mathrm{i} \cdot \mu_i' \cdot \omega \cdot \chi_i(\omega) + \sum_{j=1}^{k} A_{ij}' \cdot \chi_i(\omega) \cdot |\omega|^{\alpha_j} + \sum_{j=1}^{k} B_{ij}' \cdot \chi_i(\omega) \cdot \mathrm{sign}(\omega) \cdot |\omega|^{\alpha_j}.$$

In other words, we need to find the coefficients μ_i', A_{ij}', and B_{ij}' by applying the Least Squares method to the approximate equality

$$-\ln(\chi_i(\omega)) \cdot \chi_i(\omega) \approx$$

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

Thus, we arrive at the following algorithm.

4 A New Granular-Based Algorithm for Data Processing under Probabilistic Algorithm

Problem: reminder. We know:

- the probability distributions for $\Delta x_1, \ldots, \Delta x_n$, and
- the coefficients $c_1, \ldots, c_n$.

We want to find the probability distribution for

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

Preliminary step. We select the values $\alpha_1 < \ldots < \alpha_k$. For example, we can 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 $\alpha_3 = 2$.

We then represent the probability distribution of each random variable z by parameters μ, $A_1, \ldots, A_k$, and $B_1, \ldots, B_k$ for which

$$E[\exp(\mathrm{i} \cdot \omega \cdot z)] \approx \exp\left(\mathrm{i} \cdot \mu \cdot \omega - \sum_{j=1}^{k} A_j \cdot |\omega|^{\alpha_j} - \sum_{j=1}^{k} B_j \cdot \mathrm{sign}(\omega) \cdot |\omega|^{\alpha_j}\right).$$

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

Main algorithm. Suppose that for each i from 1 to n, we know the values μ'_i, A'_{ij}, and B'_{ij} that describe the probability distribution for Δx_i. Then, we compute the parameters μ, A_j, and B_j that correspond to Δy as follows:

$$\mu = \sum_{i=1}^{n} c_i \cdot \mu'_i;$$

$$A_j = \sum_{i=1}^{n} |c_i|^{\alpha_j} \cdot A'_{ij};$$

$$B_j = \sum_{i=1}^{n} \mathrm{sign}(c_i) \cdot |c_i|^{\alpha_j} \cdot B'_{ij}.$$

First auxiliary algorithm. Suppose that we know the probability density function $\rho_i(\Delta x_i)$, and we want to find the corresponding parameters μ'_i, A'_{ij}, and B'_{ij}. This can be done as follows.

– first, we apply the Fast Fourier Transform to compute the values of the characteristic function $\chi'_i(\omega_k)$, $k = 1, \ldots, N$;
– then, we use the Least Squares method to solve the following system of approximate equations:

$$-\ln(\chi_i(\omega_k)) \cdot \chi_i(\omega_k) \approx$$

$$-\mathrm{i} \cdot \mu'_i \cdot \omega_k \cdot \chi_i(\omega_k) + \sum_{j=1}^{k} A'_{ij} \cdot \chi_i(\omega_k) \cdot |\omega_k|^{\alpha_j} + \sum_{j=1}^{k} B'_{ij} \cdot \chi_i(\omega) \cdot \mathrm{sign}(\omega) \cdot |\omega|^{\alpha_j}.$$

Second auxiliary algorithm. Suppose that we know the values μ, A_j, and B_j corresponding to Δy, and we want to find the probability density function $\rho(\Delta y)$. For this, we do the following:

– first, for all the values ω_k on a grid, we compute the values

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

– then, we apply Fast Inverse Fourier Transform to these values, and get the desired probability density function $\rho(\Delta y)$.

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 $\alpha_3 = 2$.

Example 1: formulation of the problem. Let us assume that:

– 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 $\rho_2(\Delta x_2) = \dfrac{1}{2} \cdot \exp(-|\Delta x_2|)$,
– and $c_1 = c_2 = 1$.

We want to find the probability distribution for

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

Example 1: applying the first auxiliary algorithm. First, we apply the *first auxiliary algorithm* to describe each of these distributions in terms of the values μ'_i, A'_{ij}, and B'_{ij}. According to the first auxiliary algorithm, first, we applied the Fourier transform to compute the computed the characteristic functions. As a result, we get

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

Then, we apply the Least Squares method on the interval $[-5, 5]$. As a result, we get the following values:

$$\mu'_1 = 0, \quad A'_{11} = A_{12} = 0, \quad A'_{13} = \frac{1}{2}, \quad B'_{1j} = 0;$$

$$\mu'_2 = 0, \quad A'_{21} = -0.162, \quad A'_{22} = 1.237, \quad A'_{23} = -0.398, \quad B'_{2j} = 0.$$

Example 1: applying the main algorithm. Then, we applied the *main algorithm* and computed

$$\mu = 0, \quad A_1 = -0.162, \quad A_2 = 1.237, \quad A_3 = 0.102, \quad B_j = 0.$$

Example 1: applying the second auxiliary algorithm. To test the quality of our results, we applied the *second auxiliary algorithm* and got the (approximate) probability density function $\rho_a(\Delta y)$ for the sum.

To gauge the quality of this approximation, we also applied the current exact ($N\cdot\ln(N)$-time) algorithm, and found the exact value $\rho(\Delta y)$. The comparison between the actual probability distribution $\rho(\Delta y)$ and the approximate pdf $\rho_a(\Delta y)$ is given on Fig. 1. The corresponding mean square error $\sqrt{\int(\rho(\Delta y)-\rho_a(\Delta y))^2\,d(\Delta y)}$ is equal to 0.01.

Example 2: formulation of the problem. Let us assume that:

- both Δx_1 and Δx_2 have Laplace distributions, with $\rho_1(\Delta x_2) = \exp(-2|\Delta x_2|)$ and $\rho_1(\Delta x_2) = \frac{3}{2}\cdot\exp(-3|\Delta x_2|)$,
- and $c_1 = c_2 = 1$.

We want to find the probability distribution for

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

Example 2: applying the first auxiliary algorithm. First, we apply the *first auxiliary algorithm* to describe each of these distributions in terms of the values μ'_i, A'_{ij}, and B'_{ij}. According to the first auxiliary algorithm, first, we applied the Fourier transform to compute the computed the characteristic functions. As a result, we get

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

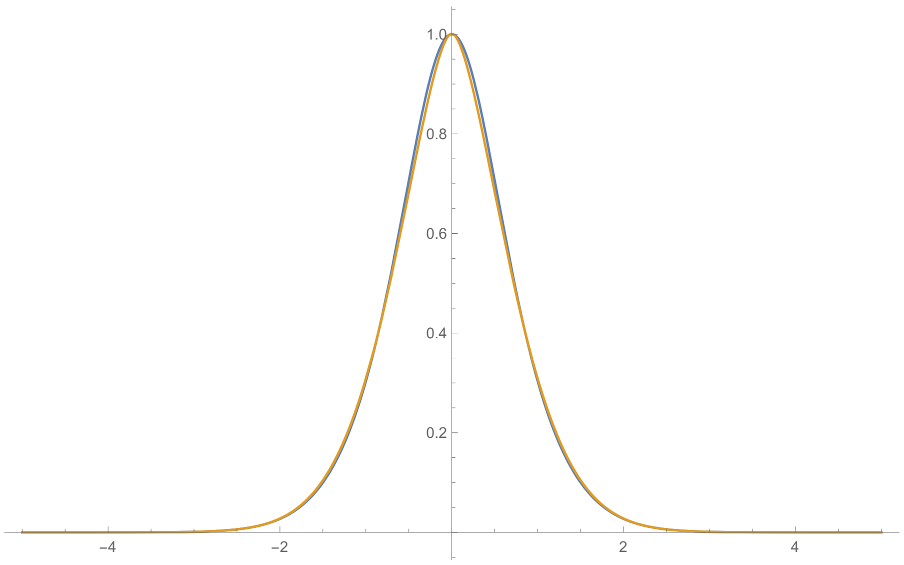

Fig. 1 How good is the proposed approximation: Example 1

Then, we apply the Least Squares method on the interval $[-5, 5]$. As a result, we get the following values:

$$\mu_1' = 0, \quad A_{11}' = 0.283, \quad A_{12}' = -0.685, \quad A_{13}' = 0.171, \quad B_{1j}' = 0;$$

$$\mu_2' = 0, \quad A_{21}' = 0.156, \quad A_{22}' = -0.327, \quad A_{23}' = 0.061, \quad B_{2j}' = 0.$$

Example 2: applying the main algorithm. Then, we applied the *main algorithm* and got

$$\mu = 0, \quad A_1 = 0.439, \quad A_2 = -1.013, \quad A_3 = 0.232, \quad B_j = 0.$$

Example 2: applying the second auxiliary algorithm. To test the quality of our results, we applied the *second auxiliary algorithm* and got the (approximate) probability density function $\rho_a(\Delta y)$ for the sum.

To gauge the quality of this approximation, we also applied the current exact algorithm, and found the exact value $\rho(\Delta y)$. The comparison between the actual probability distribution $\rho(\Delta y)$ and the approximate pdf $\rho_a(\Delta y)$ is given on Fig. 1. The corresponding mean square error $\sqrt{\int (\rho(\Delta y) - \rho_a(\Delta y))^2 \, d(\Delta y)}$ is approximately equal to 0.008 – i.e., smaller than 0.01.

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.

References

1. Cormen, Th. H., Leiserson, C. E., Rivest, R. L. Stein, C. (2009) Introduction to Algorithms, MIT Press, Cambridge, Massachusetts
2. 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
3. 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
4. Piegat, A., Landowski, M. (to appear) Solving different practical granular problems under the same system of equations, Granular Computing
5. Rabinovich, S.G. (2005) Measurement Errors and Uncertainty: Theory and Practice, Springer Verlag, Berlin
6. Sheskin, D.J. (2011) Handbook of Parametric and Nonparametric Statistical Procedures, Chapman and Hall/CRC, Boca Raton, Florida

7. Stylios, C.D., 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 $\min\limits_a \int (g(t) - a \cdot f(t))^2\,dt$. 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 $\alpha_j^{\rm opt}$ for which $\min\limits_j R(|t|^{\alpha_j}, |t|^{\alpha_{j+1}})$ is the largest is equivalent to looking for the values $\alpha_j^{\rm opt}$ 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:

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

One can check that this happens when $\alpha_j + 0.5 = 1.5 \cdot \left(\dfrac{5}{3}\right)^{(j-1)/(k-1)}$. 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 $\left(\dfrac{5}{3}\right)^{1/(k-1)}$, 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

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

for which the logarithms $\ln(\alpha_j^{\rm opt} + 0.5) = \dfrac{j-1}{k-1} \cdot \ln\left(\dfrac{5}{3}\right) = \ln(1.5)$ 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.