How to Detect (and Analyze) Independent Subsystems of a Black-Box (or Grey-Box) System

Saeid Tizpaz-Niari[1], Olga Kosheleva[2], and Vladik Kreinovich[1]
Departments of Computer Science[1] and Teacher Education[2]
University of Texas at El Paso
500 W. University, El Paso TX 79968, USA
saeid@utep.edu, olgak@utep.edu, vladik@utep.edu

Abstract

Often, we deal with black-box or grey-box systems where we can observe the overall system's behavior, but we do not have access to the system's internal structure. In many such situations, the system actually consists of two (or more) independent components: a) how can we detect this based on observed system's behavior? b) what can we learn about the independent subsystems based on the observation of the system as a whole? In this paper, we provide (partial) answers to these questions.

1 Need to Detect (and Determine) Independent Subsystems: Formulation of the Problem

Black-box and grey-box systems. Often, we only have a so-called black-box access to the system: namely, we can check how the system reacts to different inputs, but we do not know what exactly is happening inside this system. For example, we have a proprietary software, we can feed different inputs to this software and observe the results, but we do not know how exactly this result is generated.

In other cases, we have what is known as grey-box access: we have some information about the system, but not enough to find out what is happening inside the system. Such situations are also typical in biomedicine, in physics, in engineering – when we try to reverse engineer a proprietary system, etc.

Independent subsystems: how can we detect them? In many practical situations, a black-box system consists of two or more independent subsystems.

For physical systems, the overall energy of the system – which we can observe – is equal to the sum of energies of its components (which we cannot observe); see, e.g., [2, 3]. Can we detect, based on the observed energies, that the system consists of two subsystems? And if yes, can we determine the energies of subsystems?

Similarly, if a software consists of two independent components, then each observable state of the system as a whole consists of the states of the two components. Since the components are independent, the probability of each such state is equal to the product of probabilities of the corresponding states of the two components. Can we detect, based on the observed probabilities of different states, that the system consists of two subsystems? And if yes, can we determine the probabilities corresponding to subsystems?

What we do in this paper. Sometimes, the accuracy with which we measure energy or probability is very low. In such cases, we can hardly make any conclusions about the system's structure.

In this paper, we consider a frequent case when we know these values with high accuracy, so that in the first approximation, we can safely ignore the corresponding uncertainty and assume that we know the values of energies or probabilities. In such cases, we show that the detection of subsystems – and the determination of their energies or probabilities – is possible (and computationally feasible) in almost all situations.

2 Formulation of the Problem in Precise Terms and the Main Result: Case of Addition

In this section, we consider the case of addition.

Definition 1.

- *Let A and B be finite sets of real numbers, each of which has more than one element. We will call A the* set of possible value of the first subsystem *and B the* set of possible values of the second subsystem.
- *For each pair (A, B), by the* observed set*, we mean the set $A + B \stackrel{\text{def}}{=} \{a + b : a \in A, b \in B\}$ of possible sums $a + b$ when $a \in A$ and $b \in B$.*

Comment. In mathematics, the set $A + B$ is known as the *Minkowski sum* of the sets A and B.

Discussion. The problem is, given the observed set $A + B$, to reconstruct A and B.

Of course, we cannot reconstruct A and B exactly since, if we add a constant c to all the elements of A and subtract c from all elements of B, the observed set $A + B$ will remain the same. Indeed, in this case, each pair of elements $a \in A$ and $b \in B$ gets transformed into $a+c$ and $b-c$, so we have $(a+c)+(b-c) = a+b$ and thus, indeed, each sum $a + b \in A + B$ remains the same.

So, by the ability to reconstruct, we mean the ability to reconstruct modulo such an addition-subtraction.

In some cases, it is not possible to reconstruct the components: for example, for $A = \{0, 1\}$ and $B = \{0, 2\}$, the Minkowski sum $A + B = \{0, 1, 2, 3\}$ can also be represented as $\{0, 1\} + \{0, 1, 2\}$. This alternative representation has a different number of elements and thus, cannot be obtained from the original one by addition and subtraction of a constant.

What we will show, however, is that in almost all cases, reconstruction is possible.

Definition 2. *We say that the pair* (A, B) *is* sum-generic *if the following numbers are all different:*

- *all the non-zero differences* $a - a'$ *between elements of the set* A*,*
- *all the non-zero differences* $b - b'$ *between elements of the set* B*, and*
- *all the sums* $(a - a') + (b - b')$ *of these non-zero differences.*

Proposition 1. *For every two natural numbers* $n > 1$ *and* $m > 1$*, the set of all non-sum-generic pairs* (A, B) *with* n *elements in* A *and* m *elements in* B *has Lebesque measure 0.*

Discussion. In other words, almost all pairs (A, B) are sum-generic.

Proof. Indeed, the set of non-sum-generic pairs is a finite union of the sets described by equalities like $a_i - a_j = a_k - a_\ell$, $a_i - a_j = b_k - b_\ell$, etc. Each such set is a hyperplane in the $(n + m)$-dimensional space of all the pairs $(A, B) = (\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_m\})$ and thus, has measure 0. The union of finitely many sets of measure 0 is also of measure 0. So, the proposition is proven.

Proposition 2. *If* (A, B) *and* (A', B') *are sum-generic pairs for which* $A + B = A' + B'$*, then:*

- *either there exists a constant* c *for which* $A' = \{a + c : a \in A\}$ *and* $B' = \{b - c : b \in B\}$*,*
- *or there exists a constant* c *for which* $A' = \{b - c : b \in B\}$ *and* $B' = \{a + c : a \in A\}$*.*

Discussion. Thus, in the sum-generic case, we can reconstruct the components A and B from the observed set $S = A + B$ – as uniquely as possible.

Proof.

1°. Let us sort the elements of each of four sets A, B, A', and B' in decreasing order:

$$a_1 > \ldots > a_n, \quad b_1 > \ldots > b_m,$$
$$a'_1 > \ldots > a'_{n'}, \quad b'_1 > \ldots > b'_{m'}.$$

2°. The largest element of the set $A + B$ is the sum $a_1 + b_1$ of the largest element a_1 of the set A and the largest element b_1 of the set B. The next largest element of the set $A + B$ is obtained when we add the largest element of one of the sets and the second largest element of another set, so it is either

$a_1 + b_2$ or $a_2 + b_1$. These two numbers are different, since if they were equal, we would have $a_1 - a_2 = b_1 - b_2$, which contradicts to the assumption that the pairs (A, B) is sum-generic. Without losing generality, let us assume that $a_1 + b_2 > a_2 + b_1$.

Similarly, in the Minkowski sum $A' + B'$, the largest element is $a'_1 + b'_1$, and the next element is either $a'_1 + b'_2$ or $a'_2 + b'_1$. If the value $a'_2 + b'_1$ is the second largest, let us rename A' into B' and B' into A'. Thus, without losing generality, we can assume that the element $a'_1 + b'_2$ is the second largest.

Since the sets $A+B$ and $A'+B'$ coincide, this means that they have the same largest element and the same second largest element, i.e., that $a_1 + b_1 = a'_1 + b'_1$ and $a_1 + b_2 = a'_1 + b'_2$.

3°. Let us prove that for $c \stackrel{\text{def}}{=} a'_1 - a_1$, we have $a'_i = a_i + c$ and $b'_j = b_j - c$ for all i and j.

3.1°. For $i = 1$, the condition $a'_1 = a_1 + c$ follows from the definition of c.

3.2°. Let us prove the desired equality for $j = 1$.

Indeed, from the fact that $a_1 + b_1 = a'_1 + b'_1$, we conclude that $b'_1 = b_1 - (a'_1 - a_1)$, i.e., that $b'_1 = b_1 - c$.

3.3°. Let us now prove the desired equality for $j = 2$.

Indeed, the difference between the largest and the second largest elements of the set $A+B$ is equal to $(a_1+b_1)-(a_1+b_2) = b_1 - b_2$. Based on the set $A'+B'$, we conclude that the same difference is equal to $b'_1 - b'_2$. From $b_1 - b_2 = b'_2 - b'_2$, we conclude that $b'_2 = b_2 + (b'_1 - b_1) = b_2 - c$. So, the desired equality holds for $j = 2$ as well.

3.4°. Let us prove the desired equality for all i.

Since the pair (A, B) is sum-generic, all the differences $(a_i + b_j) - (a_k + b_\ell)$ corresponding to $i \neq k$ or ($i = k$ and $(j, \ell) \neq (1, 2)$) are different from the difference $b_1 - b_2$. So, the only pairs of elements from the set $A + B$ whose difference is equal to $b_1 - b_2$ are differences of the type $(a_i + b_1) - (a_i + b_2)$. There are exactly n such pairs corresponding to different values $a_i \in A$, and the sorting of the largest elements of these pairs leads to the order

$$a_1 + b_1 > a_2 + b_1 > \ldots > a_n + b_1.$$

Similarly, in the set $A' + B'$, we have exactly n' pairs whose difference is equal to $b'_1 - b'_2$, and the sorting of the largest elements of these pairs leads to the order

$$a'_1 + b'_1 > a'_2 + b'_1 > \ldots > a'_{n'} + b'_1.$$

Since $A + B = A' + B'$ and $b_1 - b_2 = b'_1 - b'_2$, these two orders must coincide, so we must have $n' = n$ and $a'_i - b'_1 = a_i - b_1$ for all i. Thus, indeed, $a'_i = a_i - (b'_1 - b_1) = a_i + c$.

3.5°. Finally, let us prove the desired equality for all j.

Indeed, since the pair (A, B) is sum-generic, all the differences $(a_i + b_j) - (a_k + b_\ell)$ corresponding to $j \neq \ell$ or ($j = \ell$ and $(i, k) \neq (1, 2)$) are different from the difference $a_1 - a_2$. So, the only pairs of elements from the set $A + B$ whose difference is equal to $a_1 - a_2$ are differences of the type $(a_1 + b_j) - (a_2 + b_j)$. There are exactly m such pairs corresponding to different values $b_j \in B$, and the sorting of the largest elements of these pairs leads to the order

$$a_1 + b_1 > a_1 + b_2 > \ldots > a_1 + b_m.$$

Similarly, in the set $A' + B'$, we have exactly m' pairs whose difference is equal to $a'_1 - a'_2$, and the sorting of the largest elements of these pairs leads to the order

$$a'_1 + b'_1 > a'_1 + b'_2 > \ldots > a'_1 + b'_{m'}.$$

Since $A + B = A' + B'$ and $a_1 - a_2 = a'_1 - a'_2$, these two orders must coincide, so we must have $m' = m$ and $a'_1 - b'_j = a_1 - b_j$ for all j. Thus, indeed, $b'_j = b_j - (a'_1 - a_1) = b_j - c$.

The proposition is proven.

Proposition 3. *There exists a quadratic-time algorithm that, given a finite set S of real numbers:*

- *checks whether this set can be represented as $S = A + B$ for some sum-generic pair (A, B), and*
- *if S can be thus represented, computes the elements of the corresponding sets A and B.*

Proof. We are given the observed set S. If this set can be represented as a Minkowski sum, we want to find the values $a_1 > \ldots > a_n$ and $b_1 > \ldots > b_m$ for which the sums $a_i + b_j$ are exactly the elements of the given set S.

In our algorithm, we will follow the steps of the previous proof.

1°. Let us first sort all s elements of the given observed set S in decreasing order:

$$e_1 > e_2 > \ldots > e_s.$$

Sorting requires time $O(s \cdot \ln(s))$; see, e.g., [1].

2°. Let us take $a_1 = e_1$ and $b_1 = 0$, then we have $a_1 + b_1 = e_1$. Let us also take $b_2 = -(e_1 - e_2)$, then $e_2 = a_1 + b_2$ and $b_1 - b_2 = e_1 - e_2$. According to the previous proof, this will work if S is the desired Minkowski sum.

3°. Let us now try all pairs (e_i, e_j) and find all the pairs for which $e_i - e_j = e_1 - e_2$. Testing all the pairs requires time $O(s^2)$. If S is the desired Minkowski sum, then, as we have shown in the previous proof, the number of such pairs is n – the number of elements in the set A – and if we sort the largest elements of these pairs in the decreasing order, we will get elements

$$u_1 = a_1 + b_1 > u_2 = a_2 + b_1 > \ldots > u_n = a_n + b_1.$$

Thus, based on these elements u_i, we get $a_i = u_i - b_1$, i.e., since we chose $b_1 = 0$, we get $a_i = u_i$.

4°. Let us now find all the pairs (e_i, e_j) for which $e_i - e_j = a_1 - a_2$. Testing all the pairs requires time $O(s^2)$. If S is the desired Minkowski sum, then, as we have shown in the previous proof, the number of such pairs is m – the number of elements in the set B – and if we sort the largest elements of these pairs in the decreasing order, we will get elements

$$v_1 = a_1 + b_1 > v_2 = a_1 + b_2 > \ldots > v_m = a_1 + b_m.$$

Thus, based on these elements v_j, we get $b_j = v_j - a_1$, i.e., since we chose $a_1 = e_1$, we get $b_j = v_j - e_1$.

5°. Finally, we form the list of all the sums $a_i + b_j$, sort it (which takes time $O(s \cdot \ln(s))$), and check that the sorted list coincides element-by-element with the original sorted list $e_1 > e_2 > \ldots$ If it does, this means that the given set can be represented as $A + B$ – and we have the desired sets A and B. If the two lists are different, this means – according to the previous proof – that the given set S cannot be represented as the Minkowski sum.

The overall computation time is equal to

$$O(s \cdot \ln(s)) + O(s^2) + O(s^2) + O(s \cdot \ln(s)) = O(s^2).$$

The proposition is proven.

First numerical example. Let us illustrate the above algorithm on a simple example when $A = \{1, 2\}$ and $B = \{1, 1.3\}$. In this case, the observed set is $S = A + B = \{2, 2.3, 3, 3.3\}$. Let us show how the above algorithm will, given this set S, reconstruct the component-related sets A and B.

1°. Sorting the elements of the set S in decreasing order leads to

$$e_1 = 3.3 > e_2 = 3 > e_3 = 2.3 > e_4 = 2.$$

2°. According to the algorithm, we then take $a_1 = e_1 = 3.3$, $b_1 = 0$, and $b_2 = -(e_1 - e_2) = -(3.3 - 3) = -0.3$.

3°. Then, we find all pairs (e_i, e_j) for which $e_i - e_j = e_1 - e_2 = 0.3$. There are exactly $n = 2$ such pairs:

- the pair $(3.3, 3)$, and
- the pair $(2.3, 2)$.

So, we conclude that $n = 2$. We then sort the largest elements of these pairs – i.e., the values 3.3 and 2.3 – in decreasing order:

$$u_1 = 3.3 > u_2 = 2.3,$$

and take $a_1 = 3.3$ and $a_2 = 2.3$.

4°. Let us now find all the pairs (e_i, e_j) for which $e_i - e_j = a_1 - a_2 = 3.3 - 2.3 = 1$. There are exactly $m = 2$ such pairs:

- the pair $(3.3, 2.3)$, and
- the pair $(3, 2)$.

So, we conclude that $m = 2$. We then sort the largest elements of these pairs – i.e., the values 3.3 and 3 – in decreasing order:

$$v_1 = 3.3 > v_2 = 3,$$

and take $b_1 = v_1 - e_1 = 3.3 - 3.3 = 0$ and $b_2 = 3 - 3.3 = -0.3$.

5°. Finally, we form all the sums $a_i + b_j$:

$$a_1 + b_1 = 3.3 + 0 = 3.3, \quad a_2 + b_1 = 2.3 + 0 = 2.3,$$

$$a_1 + b_2 = 3.3 + (-0.3) = 3, \quad a_2 + b_2 = 2.3 + (-0.3) = 2.$$

We then sort these sums into a decreasing sequence:

$$3.3 > 3 > 2.3 > 2.$$

This is exactly the given set S. Thus, this set does correspond to independent components, and we have found the sets A and B corresponding to these components.

Namely, we found the sets $A = \{3.3, 2.3\}$ and $B = \{0, -0.3\}$. If we deduce 1.3 to elements in A and add 1.3 to elements in B, we get exactly the original sets A and B.

Second numerical example. Let us illustrate the above algorithm on a example when the observed set $S = \{2, 2.7, 3, 3.3\}$ cannot be represented as the Minkowski sum.

1°. Sorting the elements of the set S in decreasing order leads to

$$e_1 = 3.3 > e_2 = 3 > e_3 = 2.7 > e_4 = 2.$$

2°. According to the algorithm, we then take $a_1 = e_1 = 3.3$, $b_1 = 0$, and $b_2 = -(e_1 - e_2) = -(3.3 - 3) = -0.3$.

3°. Then, we find all pairs (e_i, e_j) for which $e_i - e_j = e_1 - e_2 = 0.3$. There are exactly $n = 2$ such pairs:

- the pair $(3.3, 3)$, and
- the pair $(3, 2.7)$.

So, we conclude that $n = 2$. We then sort the largest elements of these pairs – i.e., the values 3.3 and 2.3 – in decreasing order:

$$u_1 = 3.3 > u_2 = 3,$$

and take $a_1 = 3.3$ and $a_2 = 3$.

4°. Let us now find all the pairs (e_i, e_j) for which $e_i - e_j = a_1 - a_2 = 3.3 - 3 = 0.3$. There are exactly $m = 2$ such pairs:

- the pair $(3.3, 3)$, and
- the pair $(3, 2.7)$.

So, we conclude that $m = 2$. We then sort the largest elements of these pairs – i.e., the values 3.3 and 3 – in decreasing order:

$$v_1 = 3.3 > v_2 = 3,$$

and take $b_1 = v_1 - e_1 = 3.3 - 3.3 = 0$ and $b_2 = 3 - 3.3 = -0.3$.

5°. Finally, we form all the sums $a_i + b_j$:

$$a_1 + b_1 = 3.3 + 0 = 3.3, \quad a_2 + b_1 = 3 + 0 = 3,$$

$$a_1 + b_2 = 3.3 + (-0.3) = 3, \quad a_2 + b_2 = 3 + (-0.3) = 2.7.$$

We then sort these sum into a decreasing sequence:

$$3.3 > 3 = 3 > 2.7.$$

This is different from the sorting of the given set S. Thus, the original set S does not correspond to independent components.

3 Generalization to Probabilistic Setting with Multiplication

In this section, we consider the case of multiplication.

Definition 3.

- *Let P and Q be finite sets of positive real numbers, each of which has more than one element, and for each of which, the sum of all its elements is equal to 1. We will call P the* set of possible probabilities of the first subsystem *and Q the* set of possible probabilities of the second subsystem.
- *For each pair (P, Q), by the* observed probabilities set*, we mean that set $P \cdot Q \stackrel{\text{def}}{=} \{p \cdot q : p \in P, q \in Q\}$ of possible products $p \cdot q$ when $p \in P$ and $q \in Q$.*

Observation: the product case can be reduced to the sum case. The logarithm of the product is equal to the sum of the logarithms. Thus, when the observed probabilities are equal to the products $p_i \cdot q_j$, the logarithms of these observed probabilities are equal to the sums $a_i + b_j$, where $a_i \stackrel{\text{def}}{=} \ln(p_i)$ and $b_j \stackrel{\text{def}}{=} \ln(q_j)$. Thus, by taking the logarithms, we can reduce this case to the case of the sum, and so get the following results.

Definition 4. *We say that the pair (P, Q) is* product-generic *if the following numbers are all different:*

- *all the ratios $p/p' \neq 1$ between elements of the set P,*
- *all the ratios $q/q' \neq 1$ between elements of the set Q, and*
- *all the products $(p/p') \cdot (q/q')$ of these ratios that are not 1.*

Proposition 4. *For every two natural numbers $n > 1$ and $m > 1$, the set of all non-product-generic pairs (P, Q) with n elements in P and m elements in Q has Lebesque measure 0.*

Discussion. In other words, almost all pairs (P, Q) are product-generic.

Proposition 5. *If (P, Q) and (P', Q') are product-generic pairs for which $P \cdot Q = P' \cdot Q'$, then:*

- *either $P' = P$ and $Q' = Q$,*
- *or $P' = Q$ and $Q' = P$.*

Discussion. If we did not have the condition that the sum of P-probabilities be equal to 1, we would have uniqueness modulo multiplication of all probabilities by a constant. However, due to this condition, in the product-generic case, we can uniquely reconstruct the components P and Q from the observed probabilities set $S = P \cdot Q$.

Proposition 6. *There exists a quadratic-time algorithm that, given a finite set S of positive real numbers:*

- *checks whether this set can be represented as $S = P \cdot Q$ for some product-generic pair (P, Q), and*
- *if S can be thus represented, computes the elements of the corresponding sets P and Q.*

Proof. In line with the general reduction, we form the set L of logarithms of the elements of S, then apply the algorithm from Proposition 3 to these logarithms, and, if the set L can be represented as a Minkowski-sum $A + B$, take $p_i = \exp(a_i)$ and $q_j = \exp(b_j)$.

We can then normalize these probabilities to take into account that $\sum p_i^{\text{act}} = \sum q_j^{\text{act}} = 1$, i.e., we should take

$$p_i^{\text{act}} = \frac{o_i}{\sum\limits_k p_k} \text{ and } q_j^{\text{act}} = \frac{q_j}{\sum\limits_\ell q_\ell}.$$

Alternatively, we can avoid computing exp and ln functions, and deal directly with the ratios and product instead of sums and differences of logarithms. Let us describe this idea in detail.

We are given the observed set S. If this set can be represented as a product, we want to find the values $p_1 > \ldots > p_n$ and $q_1 > \ldots > q_m$ for which the products $p_i \cdot q_j$ are exactly the elements of the given set S.

1°. Let us first sort all s elements of the given observed set S in decreasing order:

$$e_1 > e_2 > \ldots > e_s.$$

As we have mentioned, sorting requires time $O(s \cdot \ln(s))$.

2°. Let us take $p_1 = e_1$ and $q_1 = 1$, then we have $p_1 \cdot q_1 = e_1$. Let us also take $q_2 = e_2/e_1$, then $e_2 = p_1 \cdot q_2$ and $q_1/q_2 = e_1/e_2$.

3°. Let us now try all pairs (e_i, e_j) and find all the pairs for which $e_i/e_j = e_1/e_2$. Testing all the pairs requires time $O(s^2)$. If S is the desired product, then the number of such pairs is n – the number of elements in the set P – and if we sort the largest elements of these pairs in the decreasing order, we will get elements

$$u_1 = p_1 \cdot q_1 > u_2 = p_2 \cdot q_1 > \ldots > u_n = p_n \cdot q_1.$$

Thus, based on these elements u_i, we get $p_i = u_i/q_1$, i.e., since we chose $q_1 = 1$, we get $p_i = u_i$.

4°. Let us now find all the pairs (e_i, e_j) for which $e_i/e_j = p_1/p_2$. Testing all the pairs requires time $O(s^2)$. If S is the desired product, then, as we have shown in the previous proof, the number of such pairs is m – the number of elements in the set B – and if we sort the largest elements of these pairs in the decreasing order, we will get elements

$$v_1 = p_1 \cdot q_1 > v_2 = p_1 \cdot q_2 > \ldots > v_m = p_1 \cdot q_m.$$

Thus, based on these elements v_j, we get $q_j = v_j/p_1$, i.e., since we chose $p_1 = e_1$, we get $q_j = v_j/e_1$.

5°. Finally, we form the list of all the products $p_i \cdot q_j$, sort it (which takes time $O(s \cdot \ln(s))$), and check that the sorted list coincides element-by-element with the original sorted list $e_1 > e_2 > \ldots$ If it does, this means that the given set can be represented as $P \cdot Q$ – and we have (after normalization) the desired sets P and Q. If the two lists are different, this means that the given set S cannot be represented as the product.

Numerical example. Let us consider the case when $P = \{0.2, 0.8\}$ and $Q = \{0.3, 0.7\}$. In this case, the set of observed probabilities is $S = \{0.06, 0.24, 0.14, 0.56\}$. Let us show how the above algorithm will, given this set S, reconstruct the component-related sets A and B.

1°. Let us first sort all the elements of the given observed probabilities set in decreasing order:

$$e_1 = 0.56 > e_2 = 0.24 > e_3 = 0.14 > e_4 = 0.06.$$

2°. Let us take $p_1 = e_1 = 0.56$ and $q_1 = 1$, then we have $p_1 \cdot q_1 = e_1$. Let us also take $q_2 = e_2/e_1 = 0.24/0.56 = 3/7$, then $e_2 = p_1 \cdot q_2$ and $q_1/q_2 = e_1/e_2$.

3°. Let us now try all pairs (e_i, e_j) and find all the pairs for which $e_i/e_j = e_1/e_2 = 0.56/0.24 = 7/3$. There are exactly $n = 2$ such pairs:

- the pair $(0.56, 0.24)$ and
- the pair $(0.14, 0.06)$.

Thus, we conclude that $n = 2$. If we sort the largest elements of these pairs in the decreasing order, we will get elements

$$u_1 = e_1 = 0.56 > u_2 = e_3 = 0.14.$$

Thus, based on these elements u_i, we get $p_i = u_i/q_1$, i.e., since we chose $q_1 = 1$, we get $p_i = u_i$, i.e., $p_1 = 0.56$ and $p_2 = 0.14$.

4°. Let us now find all the pairs (e_i, e_j) for which $e_i/e_j = p_1/p_2 = 0.56/0.14 = 4$. There are exactly two such pairs:

- the pair $(0.56, 0.14)$ and
- the pair $(0.24, 0.06)$.

If we sort the largest elements of these pairs in the decreasi ng order, we will get elements

$$v_1 = 0.56 > v_2 = 0.24.$$

Thus, based on these elements v_j, we get $q_j = v_j/e_1$, i.e., $q_1 = 0.56/0.56 = 1$ and $q_2 = 0.24/0.56 = 3/7$.

5°. Finally, we form the list of all the products $p_i \cdot q_j$:

$$p_1 \cdot q_1 = 0.56 \cdot 1 = 0.56, \quad p_2 \cdot q_1 = 0.14 \cdot 1 = 0.14,$$

$$p_1 \cdot q_2 = 0.56 \cdot (3/7) = 0.24, \quad p_2 \cdot q_2 = 0.14 \cdot (3/7) = 0.06.$$

If we sort these numbers in a decreasing order, we get

$$0.56 > 0.24 > 0.14 > 0.06,$$

i.e., exactly the original sorted sequence of elements of the set S. Thus, the set S can be represented as a combination of two independent components.

To find the actual probabilities $p_i^{\rm act}$ and $q_j^{\rm act}$ of each component, we need to normalize the values p_i and q_j:

$$p_1^{\rm act} = \frac{p_1}{p_1 + p_2} = \frac{0.56}{0.56 + 0.14} = \frac{0.56}{0.7} = 0.8;$$

$$p_2^{\rm act} = \frac{p_2}{p_1 + p_2} = \frac{0.14}{0.56 + 0.14} = \frac{0.14}{0.7} = 0.2;$$

$$q_1^{\rm act} = \frac{q_1}{q_1 + q_2} = \frac{1}{1 + 3/7} = \frac{1}{10/7} = 7/10 = 0.7;$$

$$q_2^{\rm act} = \frac{q_2}{q_1 + q_2} = \frac{3/7}{1 + 3/7} = \frac{3/7}{10/7} = 3/10 = 0.3.$$

This is exactly what we needed to reconstruct.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

[1] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 2009.

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

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