EPiC Series in Computing

Volume 91, 2023, Pages 27–41

Proceedings of 38th International Conference on Computers and Their Applications

Mining for Conserved Constituents and Quantities

Yusuke Yokota[1] and James Hearne[2]

[1] Western Washington University, Bellingham, WA, U.S.A.
luxor.1922.yy@outlook.com
[2] Western Washington University, Bellingham, WA, U.S.A.
James.Hearne@wwu.edu

Abstract

Explaining the presence or absence of transformations in nature, such as chemical or elementary particle reactions, is fundamental to our thinking about nature. This paper describes a generic approach to the search for such conserved quantities. In the work that follows we formulate a generic approach to conserved such explanations by summoning techniques from Linear Algebra.

1 Invariants between Observations

It is fundamental to thinking that between transformations, or apparent transformations, something remains invariant, preserved in spite of phenomenal changes[1]. Primitively, this notion is found in the concept of object permanence, where an infant develops the understanding that objects persist between observations of them. In the history of philosophy, its first articulate statement can be found in the atomists Epicurus and Lucretius[2]. Their commitment to the view that the world consisted of unalterable, indecomposable units, naturally led them to embrace implicit conservation laws.

The mode of thought described below, the mode that guides the search for conserved quantities (and "eo ipso", conservation laws), is intended as a contribution to the logic of discovery, in its more modest sense[3]. It is important to note that although models which presuppose invariance underlying natural processes may lead only to data reduction, in practice such thinking is extraordinarily fruitful.

2 Related work

Concern with conservation laws came fairly early in the history of Machine Learning. Langley, Bradshaw and Simon approached the problem from the point of view of rule-based systems which was later extended to address particle physics explicitly [4, 5, 6]. Valdez-Perez approached the same domain by means of the simplex algorithm [7], and Zytkow approached the problem of discovering conservation laws by devising an algorithm abstracted from common scientific practice[8, 9]. Our approach differs from these approaches in being the most abstract, minimizing assumptions about the domain wherein conserved quantities might be sought. In this

respect, it falls in the tradition, initiated by Norwood Russell Hanson, of seeking a logic of discovery[10].

2.1 A Linear Algebra Interpretation of Conservation

We assume a set $\mathcal{O}$ of phenomenal objects, such as elementary particles or participants in chemical interactions. These objects could be subject to multiple conservation laws. For example, in general, elementary particles such as electrons are constrained by five conservation laws; conservation of electron-Lepton number, muon-Lepton number, tau-Lepton number, Baryon number, and strangeness number.

The input to the process we define consists of two sets of transformations $\mathcal{T}$, and $\mathcal{U}$. $\mathcal{T}$ consists of observed transformations T represented by a pair of bags, $\langle\{o_1, \cdots, o_k\}, \{o_{k+1}, \cdots, o_n\}\rangle$, where $o_i \in \mathcal{O}$. We use the notation $In(T)$ for $\{o_1, \cdots, o_k\}$ and $Out(T)$ for $\{o_{k+1}, \cdots, o_n\}$ to denote the input and output component bags of T. (Bags, rather than sets, are required because more than one instance of an object may partake as input or output to a natural transformation.) On the other hand, $\mathcal{U}$ consists of unobserved transformations T_u of such a pair representing a transformation, which has not been observed or which is known, or assumed not, to occur.

Explaining $\mathcal{T}$ and $\mathcal{U}$ by appealing to conservation principles is equivalent to positing, for each $o_i \in \mathcal{O}$ a vector v_i of conserved quantities. Transformations are licensed when the sum of the vectors associated with each $In(T)$ is equal to the sum of the vectors associated with each corresponding member of $Out(T)$. Transformations are forbidden when the sum of the vectors associated with objects in $In(T_u)$ is not equal to the sum of the vectors associated with the corresponding objects in $Out(T_u)$.

2.2 Indetermination

On this understanding there are a number of ways to determine the values of the vector v_i by solving the linear system of homogeneous equations implied by $\mathcal{T}$ against the constraints imposed by the inequalities implied by $\mathcal{U}$. Because the constraints implied by $\mathcal{T}$ do not confine the search for solutions to a single convex region, the resulting mathematics and algorithm is more complex than might at first be apparent. As a consequence, the problem of formalizing and implementing reasoning about conservation must include implicit criteria for selecting amongst the generally infinite set of vectors associated with each object o_j.

Three approaches seem plausible:

- Heuristics which direct the search in such a way that the first solution encountered answers to some general discovery principles. Such heuristics have indeed been advanced, although they are often couched in terminology that obscures the search bias they introduce [11, 12].
- A related principle, unrealized by the discovery literature, is to seek vectors that exhibit internal symmetries or maximum orthogonality from one another [13].
- Formulate the problem as an optimization problem; that is, seek vectors v_i which satisfy the equations implied by the transformations subject to some constraint such as the minimization of the vector norm or the minimization of the absolute value of the largest vector component [7].

It might seem that a scheme so simple is not actually faithful to the thinking that typically goes into identifying conserved quantities, most especially because such thinking is normally

situated in an elaborate theoretical setting. This is frequently but not always true, and has not been true in some notable cases, such as the discovery of the conservation of strangeness. Strangeness, when first posited, was simply a number whose invariance through strong interactions. Whether or not this form of data reduction has wide application, whether it is helpful in rendering large data sets more intelligible, is a matter of experience.

3 Logic

To find possible solutions, which satisfy given data transformations based on the existence of conservation laws, the algorithm introduces basic linear algebra concepts. By using the property of matrix multiplication, the algorithm can reduce the total amount of computations of searching possible solutions.

Let $\mathcal{T}$ be a set of observed transformations such that $\mathcal{T} = \{T_1, T_2, \cdots T_m\}$ where

$$T_i = \langle \{o_1, \cdots .o_k\}, \{o_{k+1}, \cdots, o_n\} \rangle$$

or

$$T_i = \langle \{In(T_i)\}, \{Out(T_i)\} \rangle$$

and o_i denotes phenomenal objects such as elementary particles. Similarly, let $\mathcal{U}$ be a set of unobserved transformations such that

$$\mathcal{U} = \{T_{u1}, T_{u2}, \cdots T_{um'}\}$$

where

$$T_{ui} = \langle \{o_1, \cdots .o_k\}, \{o_{k+1}, \cdots, o_n\} \rangle$$

or

$$T_{ui} = \langle \{In(Tu_i)\}, \{Out(Tu_i)\} \rangle$$

Since all transformations in the set $\mathcal{T}$ were observed, we can assume each $T_i \in \mathcal{T}$ must satisfy some conservation law(s) and this implies that every phenomenal object o_i also must obey the law(s). In other words, if an observed set satisfies k conservation laws, then for all phenomenal objects o_i have exactly k dimensions. Therefore, ultimately, there exists a $k \times n$ matrix X such that

$$D_o X^T = 0$$

where D_o is a $m \times n$ difference matrix, which expresses a gap of the number of phenomenal objects between $\{In(T_i)\}$ and $\{Out(T_i)\}$ for each transformation.

$$D_o = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}$$

The rows represent transformations and each column corresponds to identical phenomenal objects, so m is the number of transformations in the observed set, and n is the number of identical phenomenal objects.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kn} \end{bmatrix}$$

Each column of X corresponds to each identical phenomenal object. This X could tell us the presence of conserved quantities and which objects are essential for the reactions.

Furthermore, because of the basic property of matrix multiplication, the matrix X is constructed by k linearly independent one-dimensional vectors, $x_1, \cdots .x_k$. Thus, the goal of our algorithm is finding these one-dimensional vectors such that

$$D_o X^T = \left[\begin{array}{cccc} D_o x_1^T & D_o x_2^T & \cdots & D_o x_k^T \end{array} \right] = 0$$

and for unobserved set $\mathcal{U}$, we simply assume its reactions never happen because all of them violate some conservation laws, which means that

$$D_u X^T = \left[\begin{array}{cccc} D_u x_1^T & D_u x_2^T & \cdots & D_u x_k^T \end{array} \right] \neq 0$$

where D_u is a difference matrix for the unobserved set.

4 Algorithm

4.1 Overview

The algorithm follows the steps below in order to find a $k \times n$ matrix X such that $D_o X^T = 0$ and $D_u X^T \neq 0$.

1. Generate the Difference matrices, D_o and D_u from f_o, a collection of observed transformations and f_u, a collection of unobserved transformations respectively.

2. Finds a one-dimensional solution x_1, which satisfies both $D_o x_1^T = 0$ and $D_u x_1^T \neq 0$ at the same time.

3. Repeats the step 2 reasonable times, and the algorithm gets $x_1 \cdots x_i$, which are the candidates to construct the matrix X.

4. Finds a possible linearly independent set from the candidates. Each x_i can be a row vector of the matrix X.

4.2 Search Parameters

The behavior of the algorithm is controlled by three parameters: search_range, stage_limit, and try_limit. Based on these parameters' value and the difference matrix, the algorithm searches for elemental components of given data of some reactions

4.2.1 search_range

Although, in principle, conserved quantities may be of any magnitude, the algorithm assumes an upper bound to avoid indefinite search. For the experiments that follow, the absolute value of this upper bound is three. That is, possible entries for each conserved quantity must be restricted within the set $\{x \in \mathbb{Z} : -3 \leq x \leq 3\}$

4.2.2 stage_limit

On each iteration, the algorithm sets up an initial vector X, or a starting point of searching and tries to find one 1-D vector X, which satisfies $DX = 0$, where D is the difference matrix. The stage_limit parameter simply sets an upper bound on the number of these iterations. Depending on how many elements there are, the efficient number of the stage_limit might be different. For instance, if only four elements appear in given data, 500 iterations can be excessive.

4.2.3 try_limit

The algorithm continually updates the initial vector X until the algorithm finds a solution. The try_limit parameter restricts the number of this updating process. Updating vector X will be needed when $DX \neq 0$. At this point, the algorithm stores the old vector X in the record to avoid cycles in the search. Without this provision, it is possible for the algorithm to enter an infinite loop. In short, the try_limit parameter restricts the total number of searched points(vectors), possible solutions, and the order in which these are found.

5 Experimental Results

5.1 Preparations

To exhibit the role of three parameters; search_range, stage_limit, and try_limit, we demonstrate how these parameters apply to data with our artificial sample data.

First, we define a set of phenomenal objects $\mathcal{O}$ = {A, B, C, D, E}(the number of elements, $n = 5$) and assume that these entities are subject to two conservation laws, which implies each entity is 2-dimensional vector. We setup these vectors as following:

$$\text{A} = [1\ 1]^T,\ \text{B} = [1\ 0]^T,\ \text{C} = [2\ 0]^T,$$
$$\text{D} = [0\ 1]^T,\ \text{E} = [\text{-}1\ \text{-}1]^T$$

Thus, the algorithm's targets are following:

$$x_1 = [1\ 1\ 2\ 0\ \text{-}1]$$
$$x_2 = [1\ 0\ 0\ 1\ \text{-}1]$$

x_1 and x_2 represent the first and second entry of each element respectively. These two solutions represent the essential relationship between each element under the conservation laws.

Next, based on the solutions, we form the set of observed transformations $\mathcal{T}$ and the set of unobserved transformations $\mathcal{U}$.

Observed transformations $\mathcal{T}$:

A B $\rightarrow$ C D
C D D $\rightarrow$ A D B
E A A $\rightarrow$ B D
B B $\rightarrow$ C
D A $\rightarrow$ D D B
E B D D $\rightarrow$ D

Unobserved transformations $\mathcal{U}$:

A → C B
C C → D E
B E E → C
D A → E A
B B D → A A

For example, following our notation as mentioned, $In(T_1) = \{A, B\}$ and $Out(T_1) = \{C, D\}$. Similarly, $In(Tu_1) = \{A\}$ and $Out(Tu_1) = \{C, B\}$. Using $\mathcal{T}$ and $\mathcal{U}$ as input data, the algorithm generates the difference matrices D_o and D_u respectively.

$$D_o = \begin{array}{c} \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{pmatrix} 1 & 1 & -1 & -1 & 0 \\ -1 & -1 & 1 & 1 & 0 \\ 2 & -1 & 0 & -1 & 1 \\ 0 & 2 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array} \begin{matrix} \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{matrix}$$

$$D_u = \begin{array}{c} \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{pmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 1 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 & -1 \\ -2 & 2 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{matrix} \\ Tu_1 \\ Tu_2 \\ Tu_3 \\ Tu_4 \\ Tu_5 \end{matrix}$$

Both $D_o {x_1}^T$ and $D_o {x_2}^T$ must return $[0\ 0\ 0\ 0\ 0\ 0]^T$ and $D_u {x_1}^T$ and $D_u {x_2}^T$ never return $[0\ 0\ 0\ 0\ 0]^T$ because both x_1 and x_2 have to be subject to the conservation laws. The algorithm uses D_o and D_u in order to find possible solutions which satisfy these features.

5.2 Analysis of the Search Parameters

We set the search_range = 2, so the algorithm assumes that each entry of solutions can be -2, -1, 0, 1, or 2. Since the number of elements $n = 5$, there are $5^5 = 3125$ possible points in the search range. From Result 1, as the number of try_limits increases, the number of searched vectors(or points in the search range) also increases in the long run. However, locally, there are some exceptions. For example, with try_limit = 35, the algorithm checked 1936 points, but with try_limit = 40, the algorithm checked only 1925 points. The try_limit varies the contents of the record saving vectors that the algorithm has checked, and depending on what kind of vectors are in the record, the search direction can be changed. The worst situation occurs when x is updated, the updated x is still equal to some x in the record. In this case, the algorithm can keep updating x more smoothly and accomplish an efficient amount of search with try_limit = 35 rather than with 40.

Although higher try_limits tend to search a larger space, excessively high try_limits do not tend toward optimality. For this data set, try_limit = 15 seems to be optimal since it leads seven possible solutions with only 1574/3125 searched cases like much higher try_limits do. On the

other hand, with try_limit = 19, the algorithm found only four possible solutions even though it searched 1632 cases and the algorithm always found seven possible solutions with much higher try_limits. These results imply that there could exist a "nice" try_limit for a given data set, but try_limit must be large enough depending on the data.

search_range = 2, stage_limit = 50			
try_limit	Checked	Solutions	Time(sec)
10	1274	5	18.21
11	1398	5	20.31
12	1479	4	23.08
13	1486	5	25.36
14	1523	6	27.83
15	1574	7	30.79
16	1603	5	33.55
17	1643	6	35.16
18	1651	5	37.06
19	1632	4	39.77
20	1686	7	43.45
25	1801	6	52.45
30	1796	7	62.96
35	1936	7	79.46
40	1925	7	86.92
45	1960	7	103.91
50	1980	7	107.40

Result 1: Comparison between different try_limits

search_range = 2, try_limit = 15			
stage_limit	Checked	Solutions	Time(sec)
10	730	3	2.03
15	1023	4	4.42
20	1283	4	7.43
25	1416	5	11.43
30	1493	5	15.35
35	1533	7	21.90
40	1554	7	23.20
45	1572	7	27.32
50	1574	7	30.45
55	1579	7	35.05
60	1581	7	39.65

Result 2: Comparison between different stage_limits

The Result 2 shows that as the stage_limit increases, the number of checked cases also increases but the increasing tendency slows down and the algorithm is unable to find more than seven possible solutions after stage_limit > 35. This result implies that the algorithm can explore only a specific subset of all possible cases corresponding to the try_limit. This is because the algorithm follows two basic rules; 1) the difference matrix information (which entries should be updated) and 2) try_limit (when the algorithm should give up finding a solution on each stage). Hence, in terms of the number of searched vectors, increasing the try_limit is more

efficient than increasing the stage_limit, and the stage_limit can be adjusted based on the size of a given data set.

5.3 Possible solutions

For this experiment, we made imaginary phenomenal objects;

$$\text{A} = [1\ 1]^T,\ \text{B} = [1\ 0]^T,\ \text{C} = [2\ 0]^T,$$
$$\text{D} = [0\ 1]^T,\ \text{E} = [\text{-}1\ \text{-}1]^T$$

and these objects follow only "two" conservation laws. Also, we made data of transformations based on these objects' quantities. However, the algorithm found "seven" possible solutions using the transformations as the input data. The Result 3 shows seven possible solutions found by the Algorithm, which means that these solutions satisfy all conservation laws we assumed. Since [solution]*(-1) always satisfies the difference matrix conditions, in the real world, it is hard to distinguish between a solution and the solution*(-1) based on conservation laws only.

Possible solutions(found order)	
$\begin{bmatrix} A & B & C & D & E \end{bmatrix}$	Identity
$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	Trivial solution
$\begin{bmatrix} 1 & 0 & 0 & 1 & -1 \end{bmatrix}$	Exact solution x_2
$\begin{bmatrix} 1 & 1 & 2 & 0 & -1 \end{bmatrix}$	Exact solution x_1
$\begin{bmatrix} -1 & 0 & 0 & -1 & 1 \end{bmatrix}$	$x_2 * (-1)$
$\begin{bmatrix} -1 & -1 & -2 & 0 & 1 \end{bmatrix}$	$x_1 * (-1)$
$\begin{bmatrix} 0 & 1 & 2 & -1 & 0 \end{bmatrix}$	Unknown
$\begin{bmatrix} 0 & -1 & -2 & 1 & 0 \end{bmatrix}$	Unknown$*(-1)$

Result 3: Possible solutions found by the algorithm

The Result 3 can be significant in real applications. In this experiment, a possible solution $\begin{bmatrix} 0 & 1 & 2 & -1 & 0 \end{bmatrix}$ and its negative are not actual solutions because we assume each object is subject to only two conservation laws. However, practically, it is possible that an unknown solution indicates the existence of the unknown relationship between phenomenal objects. Thus, in practical applications, it is desirable to find as many possible solutions as possible. However, as the number of phenomenal objects increases, the running time of the algorithm also dramatically increases because a larger data set requires a combination of higher try_limit and stage_limit, and also verifying conservation laws requires matrix multiplications for each case. It is not realistic to increase stage_limit and try_limit indefinitely. At some point, we have to stop running the algorithm and analyze its intermediate output to approximate solutions for the large data set.

5.4 Elementary Particles in Physics

Here we apply the algorithm to real data from the world of elementary particle interactions(Table 1).

Input: Elementary particle reactions	
Observed reactions	Unobserved reactions
$p + p \to p + p + \pi^0$	$p \to \bar{e} + \gamma$
$p + p \to p + \pi + n$	$p \to \pi + \pi^0$
$p + \pi \to \pi + p$	$p \to \pi + \gamma$
$\bar{\pi} + p \to \bar{\pi} + p$	$p \to \pi + \pi + \bar{\pi} + \pi^0 + \pi^0$
$\bar{\pi} + p \to \pi^0 + n$	$p + p \to \bar{\Lambda} + \bar{\Lambda}$
$\bar{\pi} + p \to p + \pi + \bar{\pi} + \bar{\pi}$	
$\gamma + e \to \gamma + e$	
$e + p \to e + p$	
$\pi^0 \to \gamma + \gamma$	
$\bar{\pi} \to \mu + \bar{\nu_\mu}$	
$\pi \to \bar{\mu} + \nu_\mu$	
$\mu \to e + \nu_\mu + \bar{\nu_e}$	
$n \to p + e + \bar{\nu_e}$	
$\bar{\pi} + p \to \Lambda + K^0$	

Table 1: The set of elementary particle reactions([7])

It is known that in the interactions presented in this data set, particle interactions are constrained by five conservation laws; conservation of electron-Lepton number, muon-Lepton number, tau-Lepton number, Baryon number, and strangeness number. In other words, each particle can be represented by the five-dimensional vector as following;

A =[0 0 0 1 0]^T, B =[0 0 0 0 0]^T, C =[0 0 0 0 0]^T
D =[0 0 0 1 0]^T, E =[0 0 0 0 0]^T, F =[0 0 0 0 0]^T
G =[1 0 0 0 0]^T, H =[0 1 0 0 0]^T, I =[0 -1 0 0 0]^T
J =[0 -1 0 0 0]^T, K =[0 1 0 0 0]^T, L =[-1 0 0 0 0]^T
M =[0 0 0 1 -1]^T, N =[0 0 0 0 1]^T, O =[-1 0 0 0 0]^T
P =[0 0 0 1 1]^T

For convenience, we substitute Roman letters for the Greek symbols that are conventional in particle physics. Each alphabet member represents an identical particle. For example, 'A' is p, 'B' is π^0, ... respectively. Also, the first entry is electron-Lepton number, the second is muon-Lepton number, the third is tau-Lepton number, the fourth is Baryon number, and the fifth is Strangeness number. Combining these vectors' entries, we get possible solutions for each conservation law as following;

x_1=[0 0 0 0 0 0 1 0 0 0 0 -1 0 0]: Electron-Lepton #
x_2=[0 0 0 0 0 0 0 1 -1 -1 1 0 0 0]: Muon-Lepton #
x_3=[0 0 0 0 0 0 0 0 0 0 0 0 0 0]: Tau-Lepton #
x_4=[1 0 0 1 0 0 0 0 0 0 0 0 1 0]: Baryon #

x_5=[0 0 0 0 0 0 0 0 0 0 0 0 -1 1]: Strangeness #

Each entry represents an identical elementary particle. Each solution x_1 to x_5 constructs the solution matrix X as a row of X. Also, in this experiment, we ignore particles O and P because they do not appear in observed reactions. Since the observed set includes 14 elements, the total number of possible cases is $5^{14} = 6,103,515,625$ for search_range = |2|(The candidate of each entry is -2, -1, 0, 1, or 2). The algorithm reads Table 2, which is the modified version of Table 1. The algorithm first constructs a difference matrix D_o for observed reactions and a difference matrix D_u for unobserved reactions as we explained in the section VI.1.

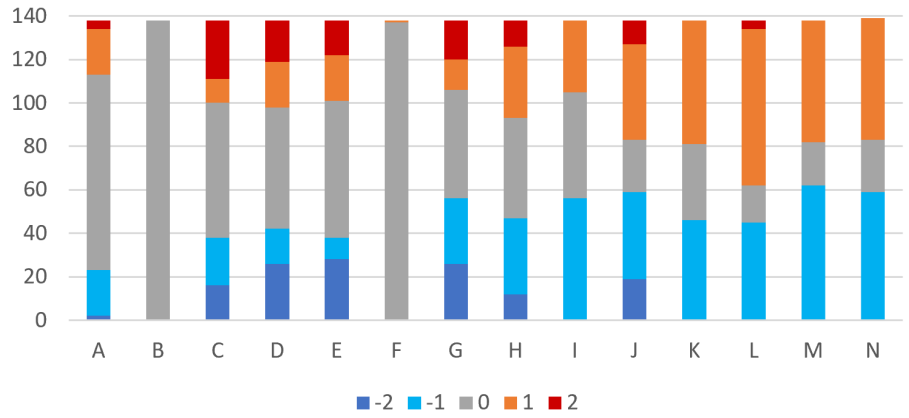
Input: Elementary particle reactions	
Observed reactions	Unobserved reactions
A A → A A B	(A → O F)ignored
A A → A C D	A → C B
A C → C A	A → C F
E A → E A	A → C C E B B
E A → B D	(A A → P P)ignored
E A → A C E E	
F G → F G	
G A → G A	
B → F F	
E → H I	
C → J K	
H → G K L	
D → A G L	
E A → M N	

Table 2: Particle reactions in Roman alphabet

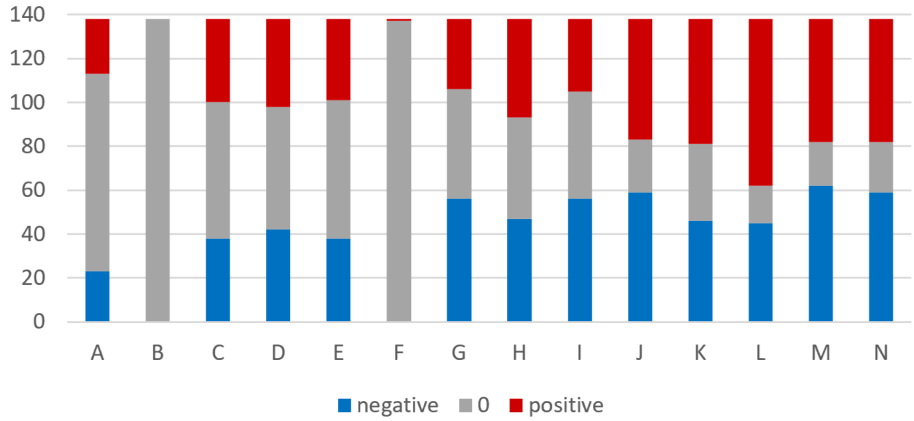
To converge on the exact solutions, the algorithm keeps recording vectors that are more than 75% close to exact solutions. For example, if a vector X passed 75% of the guiding constraints provided by the observed reactions, we assume the vector X is much closer to exact solutions rather than ones that do not. Of course this is not exactly true because we ignore the unobserved set, but as we mentioned in section 3, within limited time, directly searching the solutions is almost hopeless. The Result 4 shows the result of the first attempt with parameter conditions. At this point, the algorithm was not able to find any exact solutions.

stage_limit = 150
try_limit = 20
search_range = 2
Checked 41021 cases
Time: 9386.402 seconds
More than 75 % passed records:138
Result 4: 75% passed partially correct solutions

Here, we analyze 138 outputs, which have passed 75 % of rows of the observed difference matrix in order to gain clues for reducing search_range and finding entries of the exact solutions.



Result 5: Frequency of values of each element



Result 6: Positive vs Negative frequency

Result 5 shows the frequency of values(-2, -1, 0, 1, or 2) of each elemental particle within 138 records, which have been found by the algorithm in the Result 4. The Result 6 is a simpler version of Result 5 to check positive/negative trends of each particle. Since the algorithm updates a vector based on the difference matrix information, from the 75% passed records, we could guess the region of the search space for each particle's vector is located. Above all, if the gap between negative and positive frequency of a particle is small enough(here, say 0 or 1), we could say that the negative/positive trend of its vector seems to be "balanced," which means that the algorithm keeps searching a possible solution for the particle symmetrically with respect to the zero-vector or just searching along with the zero-vector. Thus, such a particle can be a zero vector with high probability.

Positive/Negative Trend					
o_i	$-$	**0**	$+$	Trend	Solution
A	23	90	25	$+$	[0 0 0 1 0]
B	0	138	0	Balanced	**[0 0 0 0 0]**
C	38	62	38	Balanced	**[0 0 0 0 0]**
D	42	56	40	$-$	[0 0 0 1 0]
E	38	63	37	Balanced	**[0 0 0 0 0]**
F	0	137	1	Balanced	**[0 0 0 0 0]**
G	56	50	32	$-$	[1 0 0 0 0]
H	47	46	45	$-$	[0 1 0 0 1]
I	56	49	33	$-$	[0 -1 0 0 0]
J	59	24	55	$-$	[0 -1 0 0 0]
K	46	35	57	$+$	[0 1 0 0 0]
L	45	17	76	$-$	[-1 0 0 0 0]
M	62	20	56	$+$	[0 0 0 1 -1]
N	59	23	56	$-$	[0 0 0 0 -1]

Result 7: Positive vs Negative trend

The Result 7 explains the Result 6 numerically. In this case, we could expect particle B, C, E, and F to be zero vectors and actually they are. Even though the tendency from Table 3 is not accurate enough to determine every particle's location, at least we can conclude B and F are clearly a zero vector since their frequency of zeros are extremely high. So, we assume that B = [0 0 0 0 0] and F = [0 0 0 0 0]. In addition, from Result 5, it seems that we could reduce the search_range because -2 and 2 appeared a few times for each particle's entries. Thus, the algorithm sets search_range = $|1|$. Also, instead of 75%, the algorithm tries to find 100% passed solutions that satisfy all observed reactions. The Result 8 shows the result of the second attempt with new parameter conditions. The algorithm found 12 possible solutions, which can satisfy the matrix D_o. Unfortunately, the difference matrix D_u did not work as well as we expected this time because D_u does not have enough information for most of the particles.

stage_limit = 140
try_limit = 84
search_range = 1
Checked 33837 cases
Time: 25324.840 seconds
More than 100 % passed records:12
Result 8: 100% passed solutions

In the following 12 outputs Result 9 shows, we identify solutions that can be interpreted as the identification of conserved quantities that are independently known. we have considered finding the exact solution x*(-1) is equivalent to finding the exact solution x. We will discuss the gap between experimental results and exact solutions later.

[A B C D E F G H I J K L M N]
[**0 0 0 0 0 0 0 0 0 0 0 0 0 0**]
[**0 0 0 0 0 0 1 -1 1 1 -1 -1 -1 1**]
[**0 0 0 0 0 0 -1 1 -1 -1 1 1 1 -1**]
[**0 0 0 0 0 0 1 1 -1 -1 1 -1 1 -1**]
[**0 0 0 0 0 0 -1 -1 1 1 -1 1 1 -1**]
[**1 0 1 0 -1 0 0 -1 0 1 0 -1 -1 1**]
[**0 0 0 0 0 0 1 -1 1 1 -1 -1 1 -1**]
[**-1 0 -1 0 1 0 0 1 0 -1 0 1 1 -1**]
[**0 0 0 0 0 0 -1 1 -1 -1 1 1 -1 1**]
[**0 0 0 0 0 0 1 0 0 0 0 -1 -1 1**]
[**-1 0 -1 0 1 0 0 1 0 -1 0 1 -1 1**]
[**0 0 0 0 0 0 1 -1 1 1 -1 -1 0 0**]

Result 9: Possible solutions

5.5 Discussion of the Output

1. x_3 = [0 0 0 0 0 0 0 0 0 0 0 0 0 0]: conservation of tau-Lepton number

   Coincidentally, in this reaction set, this is one of the exact solutions even though it is a trivial one.

2. Grouping of Particles

   If we extract specific possible solutions from the output and rearrange them as following, we can see some relationships between particles.

Possible Solutions													
A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	0	0	0	0	0	1	0	0	0	0	-1	-1	1
0	0	0	0	0	0	1	-1	1	1	-1	-1	0	0
0	0	0	0	0	0	1	-1	1	1	-1	-1	-1	1
0	0	0	0	0	0	-1	1	-1	-1	1	1	-1	1
0	0	0	0	0	0	-1	1	-1	-1	1	1	1	-1
0	0	0	0	0	0	-1	-1	1	1	-1	1	1	-1
0	0	0	0	0	0	1	1	-1	-1	1	-1	1	-1
0	0	0	0	0	0	1	-1	1	1	-1	-1	1	-1

Result 10: Rearranged Output

Especially, zero entries of the first and second solutions in Result 10 are significant clues to realize the relationship between particles. There are three groups; {G, L}, {H, I, J, K}, and {M, N}. Each of them is related to x_1 = [0 0 0 0 0 0 1 0 0 0 0 -1 0 0]; conservation of electron-Lepton number, x_2 = [0 0 0 0 0 0 0 1 -1 -1 1 0 0 0]; conservation of muon-Lepton number, and x_5 = [0 0 0 0 0 0 0 0 0 0 0 0 -1 1]; conservation of Strangeness

number respectively. This result implies that a possible solution could contains the key to multiple conservation laws.

3. Difficulty of Searching due to Zero Vectors

   Since zero vectors; B, C, E, and F assembled in one place, It seems that the algorithm tends to consider their adjacent particles; A and D as zero vectors too. Because of this unbalanced searching, it is hard to find $x_4 = [1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]$; conservation of Baryon number, which contains nonzero entries for A and D. To avoid this issue, we could introduce an additional row operation such as interchange of two rows of the matrix D_o in the middle of searching.

4. Efficiency

   In terms of efficiency of the algorithm, based on the difference matrix information, the algorithm only checked 33,837 cases out of $3^{14} = 4,782,969$ cases(with search_range $= |1|$) and found these useful results. If the reaction sets contain more information, the algorithm could get more accurate results directly.

6 Conclusion

The difference matrix of reactions provides the efficient search direction to find elemental vectors. By checking the trend of each element, we can reduce the search range and recognize zero vector elements. If some elements have a huge entry, this method does not work well because the required search_range can be too huge. Also, if observed reaction data provides enough information for every element, to find each elemental vector, further constraints are not required. What the definition of "enough information" is will be a further research topic. The efficient combination of try_limit and stage_limit based on statistical theories also will be a further research topic.

References

[1] Aristotle. Physics and metaphysics.

[2] Leucretius. On the nature of things.

[3] Elie Zahar. Logic of discovery or psychology of invention? British Journal of the Philosophy of Science, 34, 243-261., 1983.

[4] P. Langley, G. L. Bradshaw, and H. A Simon. Bacon.5:the discovery of conservation laws. Proceedings of the Seventh International Joint Conference on Artificial Intelligence (pp.121-126) (Vancouver, British Columbia: Morgan Kaufmann), 1981.

[5] S. Kocabas and P. Langley. Generating process explanations in nuclear astrophysics. Proceedings of the ECAI-98, Workshop on Machine Discovery (pp. 4-9). Brighton, UK, 1998.

[6] P. Langley and S. Kocabas. An integrated framework for extended discovery in particle physics. Unpublished manuscript, Institute for the Study of Learning and Expertise, Palo Alto, CA., 1998.

[7] R. E. Valdes-Perez. Algebraic reasoning about reactions: Discovery of conserved properties in particle physics. Machine Learning, 17(1):47-68, 1994.

[8] R. E. Valdes-Perez and J.M. Zytkow. Systematic generation of constituent models of particle families. Physical Review E, 54:2102-2110, 1996.

[9] Jan M. Zytkow and Herbert A. Simon. Normative systems of discovery and the logic of search. Synthese, 74, 65-90., 1988.

[10] Norwood Russell Hanson. Patterns of discovery; an inquiry into the conceptual foundations of science. Cambridge [Eng.] University Press, 1961.

[11] P. Langely. Data-driven discovery of physical laws. Cognitive Science, 5 , 31-54, 1981.

[12] Oliver Schulte. Inferring conservation laws in particle physics: A case study in the problem of induction. British Journal for the Philosophy of Science, 51, 771-806, 2000.

[13] James Hearne. Mining for conserved quantities. Proceedings of the 11th International Conference on Intelligent Systems: Emergent Technologies (July 18-20, 2003 Boston, Massachusetts), 2003.