Solving Inconsistent Systems of Linear Logical Equations

Arkadij Zakrevskij

United Institute of Informatics Problems of the National Academy of Sciences,
Minsk, Belarus

SLLE – a system of linear logical equations

Solving inconsistent SLLE

Suggested method

Fxample. Changing systems

Example. Solving the new system

Searching for short solutions

Experiments, results

SLLE – a system of linear logical equations

with m equations and n variables: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$

$$a_1^{\,1}x_1 \oplus a_1^{\,2}x_2 \oplus \ldots \oplus a_1^{\,n}x_n = y_1 \,,$$
$$a_2^{\,1}x_1 \oplus a_2^{\,2}x_2 \oplus \ldots \oplus a_2^{\,n}x_n = y_2 \,,$$
$$\ldots$$
$$a_m^{\,1}x_1 \oplus a_m^{\,2}x_2 \oplus \ldots \oplus a_m^{\,n}x_n = y_m \,,$$

$\boldsymbol{A}$ – a Boolean $(m\times n)$-matrix of coefficients,
$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ – a Boolean vector of unknowns,
$\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ – a Boolean vector of free terms.

Usually $\boldsymbol{A}$ and $\boldsymbol{y}$ are given, the value of $\boldsymbol{x}$ (called *root*), which satisfies every equation, is to be found.

An SLLE could be:

deterministic, or *defined*, having unique root, usually $m = n$;
undeterministic, or *undefined* – several roots, usually $m < n$;
overdefined, or **inconsistent** – no root, usually $m > n$

The last case shall be considered below.

Solving inconsistent SLLE

To solve an inconsistent SLLE $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$ with given $\boldsymbol{A}$ and $\boldsymbol{y}$ means to find such a value of vector $\boldsymbol{x}$, which satisfies the maximum number of equations.

Other formulation: to find a shortest *correction* vector $\boldsymbol{c}$ transforming $\boldsymbol{y}$ to a nearest to it *suitable vector* $\boldsymbol{y}^*$, for which the system becames consistent:

$$\boldsymbol{y}^* = \boldsymbol{y} \oplus \boldsymbol{c}.$$

$\boldsymbol{A}$	$\boldsymbol{y}$			$\boldsymbol{c}$	$\boldsymbol{y}^*$
11....	0	+		0	0
.1.111	1		not satisfied	1	0
.11..1	0	+		0	0
1.111.	0	+		0	0
11111.	0	+		0	0
..1.1.	0		not satisfied	1	1
.1....	0	+		0	0
.111.1	1	+		0	1
..1...	0	+		0	0
.11.1.	1	+		0	1
1.1...	0	+		0	0
...1.1	1	+		0	1

000110 $\boldsymbol{x}^*$ – optimal solution

Expectation γ of weight $w(\boldsymbol{c})$ of correction vector $\boldsymbol{c}$ for a **random** inconsistent SLLE $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$:

$$\gamma\,(m, n) = k, \quad \beta\,(m, n, k) = \sum_{i=0}^{k} C_n^{\,i}\, 2^{-m} < 1 \le \beta\,(m, n, k+1).$$

Suggested method

Checking all 2^n values of vector $\boldsymbol{x}$ to find an optimal solution is too time-consuming. The suggested method is more efficient. It begins with

canonizing system ($\boldsymbol{A}$, $\boldsymbol{y}$) by a set of n linearly independent rows and obtaining as a result other matrix $\boldsymbol{R}$ and vector $\boldsymbol{u}$ having n rows less. Then

Task 1. To find a value of vector $\boldsymbol{x}$, minimizing the weight w of vector $\boldsymbol{Ax} \oplus \boldsymbol{y}$.

is reduced to

Task 2. To find a value of vector $\boldsymbol{x}$, minimizing function $w\ (\boldsymbol{Rx} \oplus \boldsymbol{u}) + w\ (\boldsymbol{x})$.

That task is easier, because when a short solution exists, it could be found at a low *level of search* L: when only short values of vector $\boldsymbol{x}$ are checked consecutively, which have 0, 1, 2 ..., L ones.

Fxample. Changing systems

1. ***A*** and ***y*** are given.

2. *n* linerly independent rows are chosen – marked in ***b***.

3. System (***A***, ***y***) is canonized by ***b***, to (***A’***, ***y’***), which has *basic* and *rest* parts.

4. The rest part of the result serves as system (***R***, ***u***).

A	***y***	***b***	***A’***	***y’***		***R***	***u***
1 0 0 0	0	1	1 0 0 0	0		0 1 1 0	1
1 0 1 1	1	1	0 1 0 0	0		1 1 1 0	0
1 1 0 0	0	1	0 0 1 0	0		1 0 0 0	1
0 1 1 1	0	0	**0 1 1 0**	**1**		0 1 0 1	0
1 1 1 1	1	0	**1 1 1 0**	**0**		1 0 1 0	0
0 1 0 1	0	1	0 0 0 1	0	⇒	1 0 0 1	0
1 0 0 0	1	0	**1 0 0 0**	**1**		1 0 1 0	1
1 1 1 0	1	0	**0 1 0 1**	**0**		0 1 1 1	1
0 1 0 0	0	0	**1 0 1 0**	**0**		1 0 0 1	0
1 1 0 1	0	0	**1 0 0 1**	**0**		1 1 0 1	1
0 1 0 0	1	0	**1 0 1 0**	**1**		0 0 0 1	1
0 0 1 0	0	0	**0 1 1 1**	**1**			
1 1 0 1	0	0	**1 0 0 1**	**0**			
0 1 1 0	0	0	**1 1 0 1**	**1**			
0 1 0 1	1	0	**0 0 0 1**	**1**			

Example. Solving the new system

1. Solving system $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{u}$, i. e. finding a value of $\boldsymbol{x}$ which minimizes function $(\boldsymbol{Rx} \oplus \boldsymbol{u}) + w\,(\boldsymbol{x})$.

2. Obtaining correction vector $\boldsymbol{c}$ from $\boldsymbol{x}$ and $\boldsymbol{Rx} \oplus \boldsymbol{u}$.

3. Calculating suitable vector $\boldsymbol{y}^* = \boldsymbol{y} \oplus \boldsymbol{c}$, solving now consistent system $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}^*$ and finding $\boldsymbol{x}^*$.

$\boldsymbol{R}$	$\boldsymbol{u}$	$\boldsymbol{Rx} \oplus \boldsymbol{u}$	$\boldsymbol{c}$	$\boldsymbol{y}$	$\boldsymbol{y}^*$	$\boldsymbol{A}$
0 1 1 0	1	0	1	0	1	1 0 0 0
1 1 1 0	0	0	1	1	0	1 0 1 1
1 0 0 0	1	0	0	0	0	1 1 0 0
0 1 0 1	0	0	0	0	0	0 1 1 1
1 0 1 0	0	1	0	1	1	1 1 1 1
1 0 0 1	0	0	1	0	1	0 1 0 1
1 0 1 0	1	0	0	1	1	1 0 0 0
0 1 1 1	1	1	0	1	1	1 1 1 0
1 0 0 1	0	0	1	0	1	0 1 0 0
1 1 0 1	1	0	0	0	0	1 1 0 1
0 0 0 1	1	0	0	1	1	0 1 0 0
			1	0	1	0 0 1 0
			0	0	0	1 1 0 1
			0	0	0	0 1 1 0
			0	1	1	0 1 0 1

$\boldsymbol{x} = 1\ 1\ 0\ 1$

$\boldsymbol{x}^* = 1\ 1\ 1\ 0$

Searching for short solutions

Suppose a *short* correction vector $\boldsymbol{c}$ exists for which $w(\boldsymbol{c}) < \gamma$.

In that case the run-time T can be minimized by reducing the level L of search by means of

recognition: the current value of $\boldsymbol{x}$ is accepted as solution as soon as its weight $(\boldsymbol{Rx} \oplus \boldsymbol{u}) + w\,(\boldsymbol{x}) < \gamma$;

decomposition: several systems $\boldsymbol{R}\,\boldsymbol{x} = \boldsymbol{u}$ are solved in parallel instead of one ;

randomization: many systems are prepared for solving, with different randomly generated basics.

Experiments. Results

The following results were obtained for m = 1000 and n = 100, at $w(\boldsymbol{c})$ = 100, for different number (r) of systems $\boldsymbol{R}\ \boldsymbol{x} = \boldsymbol{u}$, solved in parallel.

	r = 1		r = 10		r = 100	
N	L	T	L	T	L	T
1	10	2y	3	10s	3	6m
2	12	112y	8	27d	3	6m
3	10	2y	7	4d	3	7m
4	12	112y	5	33m	4	12m
5	10	2y	5	1h	3	7m
6	14	5000y	7	3d	2	6m
7	9	69d	6	3h	4	15m
8	4	12s	4	25s	4	8m
9	6	1h	4	2m	4	9m
10	10	2y	5	52m	5	1h
11	9	69d	8	49d	4	30m
12	8	7d	6	14h	4	19m
13	9	60d	7	2d	4	28m
14	11	15y	7	10d	2	6m
15	10	2y	7	10d	2	6m
16	12	112y	7	4d	4	17m
17	13	780y	6	5h	4	18m
18	10	2y	7	7d	3	7m
19	12	112y	3	8s	3	6m
20	7	13h	6	2h	2	6m