Confidentiality for Cox model

The WebDISCO method for estimating a horizontally partitioned Cox model does not fully protect data confidentiality because predictors can potentially be identified by the central server. This vulnerability arises because data is sent for each event time. When only one event occurs during a specific time period, the predictors for that individual may become identifiable, compromising their anonymity.

To address this issue, it is crucial to ensure that there is never a single event associated with a single event time. One way to achieve this is by dividing the data into intervals where multiple patients are grouped together. For the WebDISCO method, to maintain confidentiality, an interval should either contain at least x events (five or more events, for example), or no events/censoring at all. Various methods can be used to achieve this grouping.

In the examples of the proposed methods below, data is grouped into pairs for simplicity, without considering the status (event or censoring). However, the actual implementation ensures that each group must contain at least five events (which may be changed by the user) or be entirely empty, in order to protect individual confidentiality.

Averaging

The data is ordered by time, and values are grouped and given a new time, which is the average of all values of time within a given group.

Example.

For the example, we assume we want to group data into groups of 2.

time	p_1	p_2
2	43	0
4	24	0
5	41	1
6	37	1
9	53	0
11	33	1
12	39	1
17	45	0

Before

time	p_1	p_2
3	43	0
3	24	0
5.5	41	1
5.5	37	1
10	53	0
10	33	1
14.5	39	1
14.5	45	0

After

Where p_1 and p_2 are predictors.

For this method, no communication is required between sites; all computations are performed locally in a single iteration. This method tends to generate the most intervals. A few points to note:

- Since values (event times and censored times) are grouped into sets of five in ascending order, two data points with the same time value might end up in different intervals. As a result, they could be reassigned different time values, potentially losing their equality. This also means that the original time ordering in the .csv data file may affect the resulting groups.

- The positional order of values can be altered by this method. For example, in one site, a value might be assigned a lower position, while a nearby value at another site might be increased. This positional "switching" could introduce errors. However, with large datasets, the intervals tend to be quite small, so this error might not be significant.
- If the last interval is too small (not enough values), the last two intervals will be merged.