

# Convolution operator

This operator applies a convolution filter, which is often used in electronic signal processing or in image processing to clean up wrong values like outliers. The idea behind the convolution is to correct the current value by looking at its neighbours. The number of neighbours is the size of the filter. If, for example, `SIZE=3`, the filter uses the three values before the current and three values after the current value to correct the current value. Therefore, the filter does not deliver any results for the first `SIZE` values, because it also needs additionally `SIZE` further values after the current one!

The `ATTRIBUTES` parameter is a list of all attributes where the filter should be used. Note, this is not a projection list, so that an incoming schema of "speed, direction, size" is still "speed, direction, size" after the operator. Each attribute of the schema that is not mentioned in `ATTRIBUTES` is not recognized and just copied from the input element. Each mentioned element, however, is replaced by the filtered value.

The filtered value is calculated by `FUNCTION`, which is a density function. Currently, the following functions are directly implemented:

Gaussian / Normal Distribution, where `blocked URL` is the mean and `blocked URL` the standard deviation (notice, `blocked URL` is the variance!)

`blocked URL`

Logarithmic Distribution, where `blocked URL` is the mean and `blocked URL` the standard deviation (notice, `blocked URL` is the variance!)

`blocked URL`

Uniform Distribution (where  $b-a = \text{SIZE} \cdot 2 + 1$ ):

`blocked URL`

Exponential Distribution (with `blocked URL`):

`blocked URL`

So, `FUNCTION` may have the following values: "gaussian", "uniform", "logarithmic", "exponential". Alternatively, you may add your own function, where you can use the parameter `x` to denote the value within the density function.

The `GROUP_BY` is an optional list of attributes and is used like in aggregations. For example, if the attribute "id" is added to the list of `GROUP_BY`, the convolution only considers values that come from elements with same "id". Notice, this may increase the time of waiting, because the filter needs at least  $(\text{SIZE} \cdot 2) + 1$  values for each "id".

The `OPTIONS` are used to pass the optional parameters for the standard density functions. The parameters and their defaults are as follows:

for gaussian and logarithmic:

- mean (default = 0.0)
- deviation (default = 1.0)

for exponential:

- alpha (default = 1.0)

the uniform parameters (a and b) are derived from the `SIZE` parameter.

A simple **example**: If `SIZE=3`, we look at the 6 neighbours, which are at the following indices where the current object is at index 0:

`[-3, -2, -1, 0, 1, 2, 3]`

For a simple example, we have the following values at these indices:

`[1, 2, 3, 10, 5, 6, 7]`

Thus, at this point in time, we use the values `[1, 2, 3, 10, 5, 6, 7]` to filter (correct, convolute) the value 10 (our current object). This is done as follows.

The indices `[-3, -2, -1, 0, 1, 2, 3]` are weighted by using each index as the parameter `x` for the density function  $f(x)$ . Which means, that we have the following weights for a uniform density function (where `x` is not really used and  $a-b = \text{SIZE} \cdot 2 + 1 = 7$ ):

`[1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7]`

This weights are used to calculate the percentage of each value from `[1, 2, 3, 10, 5, 6, 7]`:

`[1/7 * 1, 1/7 * 2, 1/7 * 3, 1/7 * 10, 1/7 * 5, 1/7 * 6, 1/7 * 7]`

The sum of these values is 4,857 and this is the corrected value and replaces the 10 (at index 0). Thus, in this case, the outlier of 10 is smoothed to 4,857.

Notice, this filter does not recognize any time intervals or validities!

## Parameter

- `ATTRIBUTES`: The attributes where the filter should be applied on.
- `SIZE`: The size of the filter (look above)

- **FUNCTION:** The density function that is used as a filter. Possible values are: "gaussian", "logarithmic", "uniform", "exponential" or an expression, where x is the variable that is replaced by the value.
- **GROUP\_BY:** A list of attributes that are used for grouping.
- **OPTIONS:** A map of options to configure the function: "mean", "deviation", "alpha".

## Example

### Enrich Operator

```
// applies a normal distribution filter (gaussian filter) to the speed attribute and recognizes 3 values
before and 3 values after the value that has to be filtered.
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='gaussian'}, input)

// applies a normal distribution filter (gaussian filter) to the speed attribute with mean = 0 and standard
deviation = 2
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='gaussian', OPTIONS=[['mean','0.0'],
['deviation','2.0']]}, input)

// applies a normal distribution filter (gaussian filter) to the speed attribute, but only considers values
with same id by grouping by "id"
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='gaussian', GROUP_BY=['id']}, input)

// applies a log normal distribution filter to the speed attribute
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='logarithmic'}, input)

// applies a exponential distribution filter to the speed attribute
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='exponential'}, input)

// applies a exponential distribution filter to the speed attribute
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='exponential'}, input)

// applies a uniform distribution filter to the speed attribute
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='uniform'}, input)

// applies a self-defined density function (use x to get the value)
output = CONVOLUTION({ATTRIBUTES=['speed'], SIZE=3, FUNCTION='x+2/7'}, input)
```