Preprocessing web application

List of preprocessing methods implemented in the preprocessing web-application (https://mda.tools/prep) and the releated references.

List of methods

Here we assume that the dataset is a matrix $\mathbf{X} = \{x_{ij}\}$ with n rows (objects, observations, salmples) and m columns (variables).

Savitzky-Golay

Conventional Savitzky-Golay filter for supressing noise and computing derivatives. The method has three parameters, *filter width* (`Width`), *polynomial degree* (`Polynomial`) and *derivative order* (`Derivative`). Second derivative is available only for quadratic or cubic polinomials. More details can be found in [1].

There are many ways to treat the tails when applying SG. The app implements the algorithm proposed in [2].

Normalization

Normalization is a procedure that rescales values of every row of the original dataset, $r = \{r_1, ..., r_m\}$ using a particular statistic computed for these values. There are four ways (types) of normalization implemented in the app:

- `snv` — *standard normal variate*, values of a row are mean centered and divided to standard deviation, hence it results in zero mean and unit standard deviation of values in every row.

$$r_j = \frac{r_j - m_r}{s_r}$$

- `area` — unit area normalziation, if values of a row together with x-axis form a polygon, the area of this polygon will be equal to one after unit area normalization. The area is computed as a sum of absoluted values in each row and then every value is divided to this sum. Also known as *L1 norm* normalization.

$$r_j = \frac{r_j}{\sum_{j=1}^{m} |r_j|}$$

- `length` — unit length normalziation, if a row is considered as a vector in variable (column) space, this normalization makes the Euclidean length of this vector equal to one (unit vector). The length is computed as a square root of sum of squared values in each row and then every value is divided to the length. Also known as *L2 norm* normalization.

$$r_j = \frac{r_j}{\sqrt{\sum_{j=1}^{m} r_j^2}}$$

- `var` — normalziation to unit height of a specific variable (specific column in the dataset). This is typical if you have a characteristic peak of an internal standard. In this case all values in a row are divided to the value of the specific variable (whose index, `p`, is selected in the app dialog).

$$r_j = \frac{r_j}{r_p}$$

Scaling

Scaling is a procedure similar to normalization, but is applied to values of every column (variable) of the dataset, $c = \{c_1, ..., c_n\}$. Scaling usually implies two operations, *centering* and, actually, *scaling*:

$$c_i = \frac{c_i - f_0}{f_1}$$

Here f_0 is a statistic, used for centering. In the app it can be either mean or median value of a specific column. If no centering is selected, then $f_0 = 0$.

While f_1 is a statistic used for scaling. It can be standard deviation (`sd`), inter-quartile range (difference between quartiles, `iqr`), full range (difference between max and min values, `range`) or square root of standard deviation known as Pareto scaling (`pareto`). If no scaling is selected, $f_1 = 1$.

Baseline correction

Two methods for baseline correction are implemented

Extended multiplicative scatter correction (EMSC)

EMSC is introduced in [3] and also nicely described in [4]. In this case, the original spectrum is asusmed to be a linear combination of:

- a reference spectrum
- additive baseline terms (polynomials)
- multiplicative scaling
- interference

The application estimates the influence of the first three factors, without accounting for the interference (see equation 12 in [4]). It uses mean spectrum computed for the training set as a reference spectrum. The reference spectrum is stored as a model parameter when user saves the preprocessing model and then used for preprocessing of new data.

The polynomail terms are computed assuming that the wavelengths or wavenumbers (denoted as $\tilde{\nu}$ in [4]) are evenly distributed within the normalized range of $[-1, 1]$. The number of polynomial terms to account for can be selected in the method dialog. If it is 0, then the EMSC works as conventional Multiplicative Scatter Correction method correcting only the slope and the bias without accounting for polynomial baseline terms.

Assymetric least squares (ALS)

ALS is introduced in [5, 6] and is very efficient for correcting baseline in spectral data with narrow peaks (for example to account for flourescense effect in Raman spectra). The app lets you tune two parameters.

Parameter `Lambda` in the user interface defines the power of the original λ from [4, 5]. So when you select `Lambda` as e.g. 3.5, then $\lambda = 10^{3.5}$. The larger `Lambda` the less smooth the estimated baseline curve is.

Parameter `Penalty` corresponds to p described in the baseline correction algorithm in [5]. In order to give you a possibility for selecting penalty from a wide range of values, you need to select the `Penalty range` first. For example, if the range is 0.01, then you can select one of the following penalty values: $[0.01, 0.02, 0.03, ..., 0.09]$.

Trim tails

Trimming tails is a simple procedure allowing you to remove part of the variables (columns) on the left and right sides of your dataset in case if they have no practical value (e.g., noisy spectral parts). For example you can trim 10 values on both sides from the beginning or after another preprocessing method, e.g. Savitzky-Golay, which usually affects the tails. Then the next preprocessing method will not take these values into account.

Trimming can be applied several times, each next trimming will be done on top of the previos one. When you move trimming up and down in some cases the boundaries will be reset to keep the consistency and you will have to set them again.

Spike removal

A simple algorithm to remove cosmic spikes and other similar disturbances. The method is based on median absolute deviation and modified z-scores, see details in [7]. It has two parameters — `Threshold` which sets the sensitivity of the method to distinguish between spikes and natural variation as well as `Window` size which is used to compute mean for replacing the spiked value.

References

1. A. Savitzky, M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry 1964 36 (8), 1627-1639. DOI: 10.1021/ac60214a047.
2. P A. Gorry. General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method. Analytical Chemistry 1990 62 (6), 570-573 DOI: 10.1021/ac00205a007
3. H. Martens, E. Stark. Extended multiplicative signal correction and spectral interference subtraction: New preprocessing methods for near infrared spectroscopy. Journal of Pharmaceutical and Biomedical Analysis 1991 9 (8), 625-635, DOI: 10.1016/0731-7085(91)80188-F.
4. N. K. Afseth, A. Kohler, Extended multiplicative signal correction in vibrational spectroscopy, a tutorial. Chemometrics and Intelligent Laboratory Systems 2012 117, 92-99, DOI: 10.1016/j.chemolab.2012.03.004.
5. P.H.C. Eilers. A perfect smoother. Analytical Chemistry 2003 75 (14), 3631-3636, DOI: 10.1021/ac034173t.
6. P.H.C. Eilers. Parametric Time Warping. Analytical Chemistry 2004 76 (2), 404-411, DOI: 10.1021/ac034800e.
7. D. A. Whitaker, Kevin Hayes. A simple algorithm for despiking Raman spectra. Chemometrics and Intelligent Laboratory Systems 2018 179, 82-84, DOI: ISSN 0169-7439, DOI: doi.org/10.1016/j.chemolab.2018.06.009