EIGENVECTOR
RESEARCH INCORPORATED
Research, Training, and Software

196 Hyacinth Road
Manson, WA 98831
www.Eigenvector.com

Fitting Smooth Curves Part II: Fitting with a Robust Algorithm

Neal B. Gallagher

Key words: Data fitting, smoothing penalty, basis functions, robust fitting

Introduction: Flexible fitting of smooth curves to data was discussed in previous white papers [1,2] and were based on extensions to Eilers' Perfect Smoother.[3] The tools are found in the `wsmooth` and `datafit_engine` functions in PLS_Toolbox and Solo.[4] In Part I, the `datafit_engine` function employed equality constraints and fitting to basis functions using penalty functions. In Part II (this white paper), the same objective function is used with a robust algorithm in the fitting to reduce the influence of outlier measurements in the fit.

Whittaker Smoother with Equality Constraints and Basis Functions: The following objective function is used in the `datafit_engine` function to include equality constraints and fitting to basis functions

$$O(\mathbf{z},\mathbf{a})=(\mathbf{y}-\mathbf{z})^{\mathrm{T}}\mathbf{W}_0(\mathbf{y}-\mathbf{z})+\lambda_{\mathrm{s}}\mathbf{z}^{\mathrm{T}}\mathbf{D}_{\mathrm{s}}^{\mathrm{T}}\mathbf{W}_{\mathrm{s}}\mathbf{D}_{\mathrm{s}}\mathbf{z}$$
$$+\lambda_{\mathrm{e}}(\mathbf{z}_{\mathrm{e}}-\mathbf{z})^{\mathrm{T}}\mathbf{W}_{\mathrm{e}}(\mathbf{z}_{\mathrm{e}}-\mathbf{z})+\lambda_{\mathrm{b}}(\mathbf{Pa}-\mathbf{z})^{\mathrm{T}}\mathbf{W}_{\mathrm{b}}(\mathbf{Pa}-\mathbf{z})$$

where $\mathbf{y}$ is a $N\times 1$ vector of measured data, $\mathbf{z}$ is smooth curve to be fit to the data, $\mathbf{W}_0$ is a diagonal matrix of weights (typically $0\le w_{0,n}\le 1$ for $n=1,\ldots,N$, $\mathbf{D}_{\mathrm{s}}$ is a second derivative operator (e.g., $\mathbf{D}_{\mathrm{s}}\mathbf{z}$ is the second derivative of $\mathbf{z}$), λ_{s} is a scalar penalty on the smoothing term and $\mathbf{W}_{\mathrm{s}}$ is a diagonal matrix of weights with entries $0\le w_{\mathrm{s},n}\le 1$ used to relax smoothing on selected regions of the signal. The equality constraints are given in the vector $\mathbf{z}_{\mathrm{e}}$, with corresponding diagonal matrix of weights, $\mathbf{W}_{\mathrm{e}}$ (with entries $0\le w_{\mathrm{e},n}\le 1$), and scalar penalty, λ_{e}. When equality constraints are active $\lambda_{\mathrm{e}}>0$. Elements of $\mathbf{z}_{\mathrm{e}}$ with real values and corresponding elements $w_{\mathrm{e},n}>0$ are constrained. Elements of $\mathbf{z}_{\mathrm{e}}$ with NaN values have the corresponding elements $w_{\mathrm{e},n}=0$ and are not constrained. Additionally, $\mathbf{W}_{\mathrm{b}}$ is a diagonal matrix of weights with entries $0\le w_{\mathrm{b},n}\le 1$, $\mathbf{P}$ is a NxK_{b} set of basis vectors and $\mathbf{a}$ is a set of coefficients to be estimated and λ_{b} is a scalar penalty. The corresponding estimator is

$$\begin{bmatrix}\hat{\mathbf{z}}\\ \hat{\mathbf{a}}\end{bmatrix}=\boldsymbol{\Gamma}^{-1}\left(\mathbf{W}_0\mathbf{y}+\lambda_{\mathrm{e}}\mathbf{W}_{\mathrm{e}}\mathbf{z}_{\mathrm{e}}\right)\text{ where}$$

$$\boldsymbol{\Gamma}=\begin{bmatrix}\left(\mathbf{W}_0+\lambda_{\mathrm{s}}\mathbf{D}_{\mathrm{s}}^{\mathrm{T}}\mathbf{D}_{\mathrm{s}}+\lambda_{\mathrm{e}}\mathbf{W}_{\mathrm{e}}\right) & -\lambda_{\mathrm{b}}\mathbf{W}_{\mathrm{b}}\mathbf{P}\\ -\lambda_{\mathrm{b}}\mathbf{P}^{\mathrm{T}}\mathbf{W}_{\mathrm{b}} & \lambda_{\mathrm{b}}\mathbf{P}^{\mathrm{T}}\mathbf{W}_{\mathrm{b}}\mathbf{P}\end{bmatrix}.$$

An example using robust fitting is given in PLS_Toolbox: run `datafit_engine demo` with option 3. The robust fitting is governed by the `options.trbflag` input ("top or bottom" flag) that allows fitting to the 'middle' of the data cloud or fitting to the 'bottom' or 'top' when baselining the data.

In this example, `plsdata` available in PLS_Toolbox and Solo[4] includes temperature versus sample point (time) for a process. Figure 1 shows the temperature on Thermocouple 10 (blue) and smoothed versions for $\mathbf{W}_0=\mathbf{I}$ and λ_{s} = 1, 10, 10^3 and 10^4 (smoothing penalty increasing) without robust fitting. Figure 2 shows the same data with a robust fit through the middle of the data cloud. As expected, smoothing increases as the smoothing penalty increases and the robust fit is less influenced by the strong dips at Sample Points 29, 51 and 73.

Another interpretation of the smoothed fit is that it can be used to characterize the long-term trend. For example, using robust fitting (trbflag = 'middle') and increasing the smoothing penalty to $\lambda_{\mathrm{s}}=10^5$ generates a smooth "quadradic" fit to the data, **z**, as shown in the top plot of Figure 3. As expected, the fit in Figure 3 is significantly smoother than the fits shown in Figure 2 due to a larger fit penalty. Also, as before, the results in Figure 3 are not influenced by the strong dips in the data series. The short-term trend is given by **yb** = **y** – **z** [**yb** and **z** are outputs `yb` and `z` respectively from `datafit_engine`] as shown in the bottom plot of Figure 3 (red). The short-term trend (red) is compared to the mean-centered data (blue).

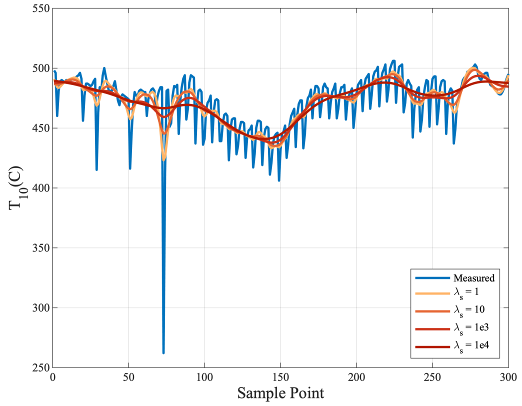

Figure 1: Measured Thermocouple 10 (blue) and smoothed data $\lambda_s > 0$. Robust fitting was not used: trbflag = 'none'.

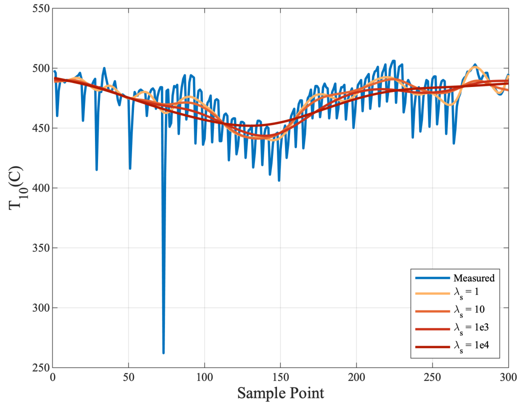

Figure 2: Measured Thermocouple 10 (blue) and smoothed data $\lambda_s > 0$. Robust fitting was used: trbflag = 'middle'.

Robust fitting is generally governed by the optional tolerance input `tol` (`options.tol`). In the above this was input manually to `tol` = 2. In the iterative robust algorithm, fit residuals > `tol` at each iteration are considered large and corresponding elements of $\mathbf{W}_0$ are set to a small number thus reducing their influence on the fit.

If not given, `tol` is determined automatically as the mean absolute deviation of the difference between the measured signal, **y**, and a Savitzky-Golay fit using a first order polynomial with a moving window width of three.[5] Robust fitting can then governed using the `tolfac` input (`options.tolfac`) where residuals > `tol*tolfac` would be considered large with respect to the fitting algorithm.

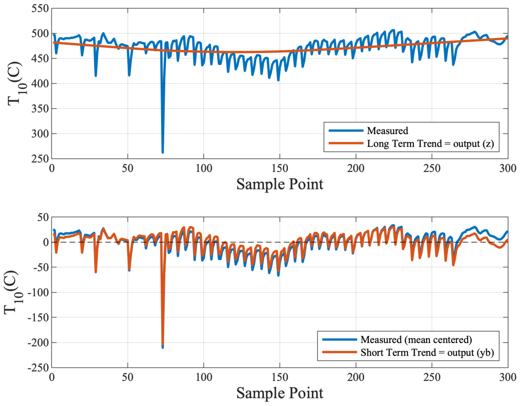

Figure 3: (top) Measured Thermocouple 10 (blue) and smoothed data $\lambda_s = 10^5$ i.e., the long-term trend z (red). Robust fitting was used: trbflag = 'middle'. (bottom) Mean-centered data (blue) compared to the short-term trend given by yb = y – z.

Conclusions: Smoothing is a useful tool for providing interpretable trends and processing of time-series data. The additional flexibility of robust fitting (shown here), and equality constraints basis functions (shown previously [2]) gives the data analyst a powerful set of tools for fitting data.

References:

[1] Gallagher, NB, "Whittaker Smoother," white paper Eigenvector Research, Inc., www.eigenvector.com.

[2] Gallagher, NB, O'Sullivan, D, "Fitting Smooth Curves Part I: Fitting with Equality Constraints and Basis Functions," white paper Eigenvector Research, Inc., www.eigenvector.com.

[3] Eilers, PHC, "A Perfect Smoother," *Anal. Chem.* 2003, **75**, 3631-3636.

[4] PLS_Toolbox and Solo. Eigenvector Research, Inc., Manson, WA USA 98831; software available at www.eigenvector.com.

[5] Savitzky, A, Golay, MJE, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Anal. Chem.* 1964, **36**(8), 1627-1639.