**Dokumentation Algorithmus «Drift Detection In Presumed Non Drifting Data»**

Wir wollen den Datendrift während der Injektion der Negativprobe durch eine Funktion approximieren. Dazu wählen wir die Langmuir Funktion

,

wobei

* : der Langmuir-Faktor ist;
* : die Konzentration ist; und
* : der Zeitpunkt des Reaktionsbeginn ist.

Für unser Fitting fügen fassen wir die uns beiden unbekannten Parameter und im Parameter zusammen und führen einen neuen Parameter ein, der die «Höhe» der Funktion steuern soll. Unsere Modelfunktion ist also

.

So beschreibt der Parameter weiterhin den Reaktionsbeginn. Der Parameter beschreibt die Geschwindigkeit, mit der das Maximum erreicht wird und genau dieses Maximum. Wir setzen für die Parameter die Bedingungen und .

Für das Fitting der Funktion gehen wir wie folgt vor: Zuerst bestimmen wir mit dem *BEAST* *(Bayesian estimator of Abrupt Change/changepoint, Seasonality, and Trend*)[1] Algorithmus drei mögliche Zeitpunkte für den Reaktionsbeginn. Für jeden dieser möglichen Zeitpunkte versuchen wir durch Optimierung der Parameter und das Bestimmtheitsmass der Funktion zu den Originaldaten zu minimieren. Der mögliche Reaktionszeitpunkt, für welchen sich die beste Optimierung (für die Beschreibung der Methode siehe weiter untern) erreichen lässt, wählen wir als unseren Zeitpunkt für den Reaktionsbeginn. Die bereits für diesen Reaktionsbeginn optimierten Parameter und übernehmen wir.

Als Methode für die Optimierung der beiden Parameter wählen wir *Non-linear least squares* und legen unsere Anfangswerte als und fest.

**MATLAB code**

ft = fittype('b\*(1 - exp(-a \* t))', independent = 't', coefficients = {'a', 'b'});

opts = fitoptions(Method = 'NonlinearLeastSquares', StartPoint = [0, 0], Lower = [0, 0]);

**Bestimmung des Reaktionsbeginn**

Für die Bestimmung des Reaktionsbeginn nutzen wir den *BEAST* Algorithmus. Methoden, den Anfangspunkt mit der Ableitung (MATLAB Funktion diff und gradient) zu bestimmen, scheiterten daran, dass sie den Anstieg zu spät erkennen (erst ab einem Schwellenwert, welcher über dem Grundrauschen liegen muss) selbst bei Glättung der Daten. Die MATLAB Funktion findchangepts aus der Signal Processing Toolbox versucht die Daten in zweit Teile zu unterteilen, welche, je nach Wahl, welche in sich einen möglichst ähnlichen Mittelwert, Standardabweichung oder Quadratisches Mittel haben sollen. Die Funktion ist ebenfalls nicht in der Lage den Zeitpunkt des Reaktionsbeginn richtig zu erkennen.

Daher haben wir uns für die Nutzung des *BEAST* Algorithmus entschieden. Dieser detektiert verschiedene mögliche «Changepoints» und gibt uns für jeden von diesen eine Wahrscheinlichkeit, mit welcher er vermutet es sei der relevante. In allen getesteten Fällen war *BEAST* in der Lage den Zeitpunkt des Reaktionsbeginn als Changepoint zu erkennen, jedoch nicht immer stufte es ihn als den relevantesten (den mit der höchsten Wahrscheinlichkeit) ein.

Wir haben uns daher entschieden jeweils die drei wahrscheinlichsten «Changepoints» von BEAST zu nehmen und für alle drei einen Fit durchzuführen. Anschliessend berechnen wir für jeden «Changepoint» einen Score aus dem Bestimmtheitsmass für den Fit , der BEAST Wahrscheinlichkeit und der Distanz zum ersten Drittel .

Die Gewichtung der Komponenten wurde so gewählt, dass alle drei Faktoren einen ähnlichen Einfluss auf den Score haben, wobei das Bestimmtheitsmass im Schnitt den grössten und die Distantz zum ersten Drittel den kleinsten Einfluss hat.

**Quellenverzeichnis**

[1] Zhao et al. (2019). Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm. *Remote Sensing of Environment, 232, 111181. doi.org:10.1016/j.rse.2019.04.034*