



MATLAB für Naturwissenschaftler

9. Lineare und nichtlineare Regression

Till Biskup

Lehrstuhl für Physikalische Chemie und Didaktik
Universität des Saarlandes



Motivation

Allgemeines zur Kurvenanpassung

Lineare Regression

Nichtlineare Regression



Daten: Wahrung der empirischen Wissenschaften

- ▶ Grundlage und Ausgangspunkt empirischer Wissenschaft
 - Daten sind nicht notwendigerweise „offensichtlich“.
 - Messung zur „Aufnahme“ von Daten

- ▶ Daten iberdauern, Interpretationen ndern sich
 - Daten nach bestem Wissen und Gewissen aufnehmen
 - Rohdaten *niemals* wegwerfen

- ▶ Verantwortung des Wissenschaftlers
 - Saubere Datenaufnahme und -dokumentation
 - Nachvollziehbarkeit der Datenaufnahme und -verarbeitung
 - Reproduzierbarkeit der Ergebnisse



Daten verarbeiten: „Ein weites Feld“

- ▶ Abhängig von der jeweiligen Fragestellung
- ▶ Meist nichttrivial und mitunter aufwendig
- ▶ Oft Anpassung von Modellen an die Daten
 - Regressionsanalyse

Konkretes Beispiel: Fluoreszenz-Versuch aus dem PCG

- ▶ Schritte auf die konkrete Fragestellung zugeschnitten
 - Zugriff auf die eingelesenen Daten
 - Umrechnung von Werten
 - Regression und Anpassung von Kurven



Warum Kurvenanpassung?

- ▶ Bestimmte Eigenschaften eines Datensatzes extrahieren
 - Z.B. das Maximum oder ein Wendepunkt
- ▶ Kurven (und Linien) als Hilfe für den Betrachter
 - Helfen oftmals beim Hervorheben von Eigenschaften.
 - Haben in der Regel keine physikalische Bedeutung.
- ▶ Beschreibung der Daten durch ein (einfacheres) Modell
 - Die Anpassung liefert die Parameter des Modells.
 - Das Modell ist i.d.R. einfacher als die Realität.
- ▶ Umsetzungstabelle finden
 - Für die Abhängigkeit zweier physikalischer Eigenschaften.
 - Wenn es keinen (einfachen) Zusammenhang gibt.



Wichtiger Hinweis

Verwechseln Sie nie Kurvenanpassung mit Physik.

- ▶ Es gibt nicht immer einen physikalischen Hintergrund
 - Hängt von der jeweiligen Fragestellung ab.
- ▶ Es gibt immer einen mathematischen Hintergrund
 - Eine Kurvenanpassung wird immer Ergebnisse liefern.
 - Diese Ergebnisse sind nicht immer (physikalisch) sinnvoll.



Ein bisschen Mathematik

- ▶ Kurvenanpassung
 - Mathematisches Modell zur Beschreibung der Daten
 - Sollte so nahe wie möglich an den Daten sein.
- ▶ Was ist „so nahe wie möglich“?
 - Wir brauchen ein Gütekriterium.
 - Häufig wird der quadratische Mittelwert der Abweichungen von Daten und Modell verwendet.
- ☞ Kurvenanpassung
 - Minimierung des quadratischen Mittelwerts der Abweichungen von Daten und Modell
 - Alternative:
Minimierung der Summe der Quadrate der Abweichungen



Ein bisschen Mathematik

Gegeben seien Daten y_k mit $k = 1 \dots M$ gemessen als Funktion einer unabhängigen Variable x an den Punkten x_k .

Die Funktion f mit den Parametern $a_i, i = 0 \dots n$ soll diese Daten bestmöglich beschreiben:

$$y_k \approx f(a_0, \dots, a_n, x_k)$$

Minimiert werden muss also die Abweichung von Funktion und Daten, hier die Summe der Quadrate der Abweichungen:

$$r = \sum_k (y_k - f(a_0, \dots, a_n, x_k))^2$$



Ein bisschen Mathematik

Das Minimum lässt sich dadurch finden, dass die partiellen Ableitungen von r nach allen Parametern a_i zu Null gesetzt werden:

$$\frac{\partial r}{\partial a_0} = 0, \quad \frac{\partial r}{\partial a_1} = 0, \quad \dots, \quad \frac{\partial r}{\partial a_n} = 0$$

Ob und wie diese partiellen Ableitungen berechnet werden können, hängt von der Form der Funktion f ab.

👉 Je mehr Parameter, desto besser die Anpassung.



Zwei Arten von Minimierungsproblemen

- ▶ Minimierungsprobleme mit eindeutigen Lösungen
 - Polynome
 - Funktionen linearer Parameter

- ▶ Minimierungsprobleme ohne eindeutige Lösung
 - Beliebige Funktionen oft mit vielen Parametern
 - Multidimensionaler Parameterraum für die Minimierung
 - Kein eindeutiger Weg der Minimierung
 - Oftmals lokale und globale Minima ununterscheidbar

- ☞ Wann immer möglich Polynome oder Funktionen linearer Parameter verwenden.



MATLAB-Routinen zur Kurvenanpassung

- ▶ Polynome

- `polyfit`, `polyval`

- ▶ Lineare Gleichungssysteme

- `\`, `lsq`

- ▶ Allgemeine nichtlineare Kurvenanpassungen

- `fminsearch`
- (kommerzielle) Toolboxes

- ☛ MATLAB stellt diverse Möglichkeiten zur Verfügung.



Lineare Regression

- ▶ Offensichtliche Ausreißer nicht berücksichtigen
 - Kurz darlegen, warum ein Datenpunkt als „Ausreißer“ betrachtet und nicht berücksichtigt wurde.
- ▶ Korrekte Formel verwenden
 - Es ist ein fundamentaler Unterschied, ob der y -Achsenabschnitt mit angepasst wird oder nicht.

Zur Beachtung

$$y = ax \quad \neq \quad y = ax + b$$
$$y = ax + \text{const} \quad \neq \quad y = ax + b$$



Lineare Regression mit zwei Parametern

- ▶ Funktion: $y = a \cdot x + b$
- ▶ MATLAB-Funktion: `polyfit`, `polyval`

Listing 1: Lineare Regression mit zwei Parametern

```
1 % Assume data x,y
2 % Fit polynomial of first order, f(x)=y=a*x+b
3 coefficients = polyfit(x,y,1);
4
5 % Get regression curve with calculated coefficients
6 regression   = polyval(coefficients,x);
```

- ▶ Anmerkungen
 - `polyfit`, `polyval` auch für Polynome n -ter Ordnung



Lineare Regression durch den Ursprung

- ▶ Funktion: $y = a \cdot x$
- ▶ MATLAB-Funktion: `lscov` oder `\`

Listing 2: Lineare Regression durch den Ursprung

```
1 % Assume data x,y
2 % Use system of linear equations, A*m = B => y = a*x
3 % Solve using "lscov" to get slope
4 a = lscov(x(:),y(:));
5
6 % Second parameter gives error estimate
7 [a,sa] = lscov(x(:),y(:));
```

- ▶ Anmerkungen
 - `lscov` ist empfindlich auf die Dimension der Vektoren.
 - `x(:), y(:)` umgeht das Problem.



Lineare Regression mit festem y -Achsenabschnitt

- ▶ Funktion: $y = a \cdot x + \text{const}$
- ▶ MATLAB-Funktion: `lscov` oder `\`

Listing 3: Lineare Regression mit festem y -Achsenabschnitt

```
1 % Assume data x,y
2 % Use system of linear equations, A*m = B => y = a*x+const
3 % Solve using "lscov" to get slope
4 a = lscov(x(:),y(:)-const);
5
6 % Second parameter gives error estimate
7 [a,sa] = lscov(x(:),y(:)-const);
```

- ▶ Anmerkungen
 - Festen y -Achsenabschnitt von den Daten y abziehen



Nichtlineare Regression beliebiger Funktionen

- ▶ Funktion: $y = a \cdot (1 - 10^{-b \cdot x})$
- ▶ MATLAB-Funktion: `fminsearch`

Listing 4: Nichtlineare Kurvenanpassung

```
1 % Assume data x,y
2 % Set starting values for coefficients c
3 c = [1000 5e-4];
4
5 % Define model to fit the data, with vector of coefficients c
6 % Model: y = c(1)*(1-10^(-c(2)*x))
7 model = @(c)c(1).*(1-10.^(-c(2).*x));
8
9 % Define fit function as sum of residual least squares
10 fitfun = @(c)sum((y-model(c)).^2);
11
12 % Get coefficients using fminsearch
13 coeff = fminsearch(fitfun,c);
```




Anmerkungen zur Funktion `fminsearch`

- ▶ Verlangt als Funktion die zu minimierende Funktion
 - Meist die Summe der Quadrate der Abweichungen
 - Definition zweier Funktionen: Modell und Fitfunktion
- ▶ Gradientenfreie Methode
 - Nelder-Mead-Simplex-Algorithmus
 - Nicht für alle Fälle geeignet
- ▶ Vorteil
 - In MATLAB ohne zusätzliche Toolboxen verfügbar
 - Für viele einfache Probleme ausreichend
- ☛ Für „ernsthafte“ Kurvenanpassungen mit MATLAB führt kein Weg an der Optimization Toolbox vorbei.



...Zeit für eigene praktische Arbeit...

Vorschau: **Projekt: Vorstellung und Pflichtenheft**

- ▶ Daten importieren
- ▶ Daten verarbeiten
- ▶ Daten darstellen