Tag 8: Modellbildung

Version vom 6.10.2014

A) KURVENANPASSUNG

Kurvenanpassung dient dazu, Messdaten durch eine Kurve - also eine mathematisch beschreibbare Funktion - zu beschreiben. Die Idee dabei ist, dass die Messdaten systematisch von einem vorgegebenen Parameterabhängen, aber statistisch um die zu erwartenden Werte streuen.

* Graphisch trägt man die Werte des Parameters (z.B. Zeit, Düngemittelmenge, Sonnenscheindauer etc) auf der x-Achse auf, die Messwerte und die theoretischen Werte in Abhängigkeit von diesem Parameter auf der y-Achse.
* Häufig werden dabei die Messwerte als Punkte (oder andere Symbole) dargestellt, die theoretischen Werte als durchgezogene Kurve.
* Diese Darstellung hilft dabei, die Idee der Interpolation zu verstehen: Durch sie lässt sich abschätzen, welche Messwerte zu erwarten gewesen wären, wenn man zwischen den gewählten Parameterwerten weitere Messpunkte ermittelt hätte.
* Außerdem ermöglicht Kurvenanpassung auch Extrapolation, die Vorhersage, wie sich die Messwerte außerhalb des getesteten Parameterbereichs verhalten.

Um eine Funktion zu finden, die die Abhängigkeit der vorhandenen Messdaten vom variierten Parameter möglichst gut beschreibt, möchte man eine Kurve finden, die möglichst nah an den Messdaten liegt. Hierfür (oder generell wenn man versucht, Messdaten durch eine Bildungsregel (also ein Modell) zu beschreiben), braucht man ein Maß dafür, wie gut das Modell die Daten widerspiegelt. Dadurch kann man verschiedene Modelle miteinander vergleichen und sich für das am besten passende entscheiden. Ein Standardverfahren, um die optimale Funktion (bzw. das optimale Modell) zu bestimmen, ist die mittlere quadratische Abweichung zwischen den Messwerten und den geschätzten Werten zu minimieren. (Prinzipiell gibt es aber noch viele andere Möglichkeiten, die Güte eines Modells zu messen.)

Hierzu wird zunächst für jeden Datenpunkt das Residuum bestimmt, also die Differenz zwischen dem empirisch ermittelten Wert yi und der Schätzung für diesen Wert (Schätzwerte macht man gerne mit einem Dach kenntlich):



Die mittlere quadratische Abweichung zwischen allen Datenpunkten und der Schätzung wird ermittelt als Summe der quadrierten Residuen, geteilt durch die Anzahl der Datenpunkte n:

 

* Was bewirkt das Quadrat?

 In diesem Kurs benutzen wir Polynome der Form

*p(x)=p1xn+p2xn-1+...pnx+pn+1*

 um den empirischen Daten eine Funktion anzupassen.

* Wie sieht in dieser Darstellung eine Gerade mit Steigung 2.5 und y-Achsenabschnitt 7 aus?
* Natürlich lassen sich nicht alle Abhängigkeiten durch Polynome adäquat darstellen. Welche Gegenbeispiele fallen Ihnen ein?

**Matlab:**

In Matlab stehen für die Kurvenanpassung eines Polynoms an Messdaten die Befehle *polyfit* und *polyval* zur Verfügung:

*[p,S]=polyfit(x,y,n)*

* Rückgabe *p=[p1 p2... pn pn+1]* gibt diejenigen Koeffizienten für die Gleichung *p(x)=p1xn+p2xn-1+...pnx+pn+1*

zurück, die den quadratischen Fehler zwischen Messwerten und geschätzten Werten minimiert.

* *S* ist eine Struktur, die *polyval* als Eingabe braucht .

*y\_schaetz = polyval(p,x,S)* bekommt als Eingaben

* den von *polyfit* zurückgegebenen Vektor *p*
* die von *polyfit* zurückgegebene Struktur *S*,
* sowie einen Vektor von *x*-Werten.

Für diese *x* -Werte berechnet *polyval* die gemäß der gefundenen Gleichung vorhergesagten y-Werte (die x Werte werden in die Gleichung eingesetzt).

*[y\_schaetz,delta] = polyval(p,x,S)* liefert zusätzlich das Ausgabeargument

* *delta* als ein Maß dafür, wie sicher diese Schätzung ist. Im Bereich *y\_schaetz-delta*bis *y\_schaetz+delta* liegen 50% der Messpunkte.

**Aufgaben:**

Ich möchte Sie bitten, zunächst einmal für einen einfachen Fall Kurvenanpassung “von Hand” vorzunehmen, damit Sie die Grundidee besser verstehen:

T8A1) Als Vorbereitung beschäftigen wir uns zuerst mal mit dem Fehlermaß, der mittleren quadratischen Abweichung: Schreiben Sie eine Funktion, die einen Vektor gemessener und einen gleichlangen Vektor geschätzter y-Werte bekommt und den mittleren quadratische Fehler zurückgibt.

\*T8A2) Erweitern Sie Ihre Funktion so, dass Sie auch für eine Matrix von Messwerten (bei der jedem x-Wert mehrere y-Werte zugeordnet sind) und einen Vektor der geschätzten y-Werte (für jeden x-Wert  einer) den mittleren quadratischen Fehler berechnet.

T8A3) Jetzt zur eigentlichen Kurvenanpassung: Wenn man Nervenzellen mit Strom reizt, reagieren sie darauf mit einer Folge von Aktionspotentialen.  Stellen Sie sich ein Neuron vor, das auf einen Stromimpuls von 1 nA mit einer Frequenz von 32 Sp/s reagiert, bei 2 nA produziert es 60 Sp/s.

* Welche Antwortfrequenz erwarten Sie bei einer Strominjektion von 1.5 nA?
* Wie lautet die lineare Gleichung für die Gerade, die beide Punkte verbindet?
* Plotten Sie die Gerade und die beiden Datenpunkte

T8A4) Sie führen jetzt eine weitere Messung durch und stellen fest, dass bei 1.5 nA Strominjektion 44 Sp/s erzeugt werden.

* Erweitern Sie Ihren plot um den neuen Datenpunkt.
* Berechnen Sie den quadratischen vertikalen Abstand des Datenpunkts von der Geraden.
* Schätzen Sie "per Auge" eine Gerade, die alle drei Punkte gut annähert und plotten sie zusammen mit den Datenpunkten.
* Berechnen Sie Ihren mittleren quadratischen Fehler.
* Benutzen Sie jetzt den Befehl *polyfit*, um die beste lineare Nährung zu finden.
* Berechnen Sie mit *polyval* den geschätzten Datenpunkt und den Fehler
* Plotten Sie die von Matlab berechnete Kurve und deren *delta*-Werte als Fehlerbalken gemeinsam mit den Datenpunkten in eine Abbildung.
* Wie gut war Ihre Schätzung?
* Eine alternative Möglichkeit zur Fehlerabschätzung ist der von *[p,S]=polyfit(x,y,n)*zurückgegebene Wert *S.normr*. Dieser Wert gibt den Betrag (euklidische Norm) der Residuen an. (Man kann ihn sich einfach z.B. mit *residuen\_betrag=S.normr* anzeigen lassen).

\*T8A5) Eigentlich braucht man die Funktion *polyval* nicht unbedingt für unsere Anwendung. Schreiben Sie eine Funktion mit

* Eingabeargument: Vektor *p* der Koeffizienten und
* Eingabeargument: *x*-Vektor
* Rückgabeargument: der resultierende Vektor der Schätzwerte
* Testen Sie am Beispiel der letzten Aufgabe, ob ihre Schätzungen denjenigen der Funktion *polyval* entsprechen.

T8A6) Um das Neuron besser zu charakterisieren wurde eine Messreihe mit Stromreizen zwischen 0 und 3.5nA durchgeführt und jeweils 10 mal die Antwortrate gemessen. Die Reize sind unter  [[reizT8.mat],](http://www.compneuro.uni-oldenburg.de/download/kurs08/tag08/reizT8.mat) die resultierenden Raten unter [[ratenT8.mat]](http://www.compneuro.uni-oldenburg.de/download/kurs08/tag08/ratenT8.mat)abgespeichert.

* Laden Sie Reize und Raten ein und tragen sie die Raten (Antworten) als einzelne Datenpunkte gegen die Reizstärken auf.
* Finden Sie die beste lineare Nährung für die Daten und plotten Sie diese dazu.
* Extrapolation: Welche Rate erwarten Sie in Antwort auf einen 5nA starken Reiz? Welche bei -2nA?

T8A7) Um unsere Extrapolation zu prüfen wurde die Messreihe erweitert Unter [[allereizeT8.mat]](http://www.compneuro.uni-oldenburg.de/download/kurs08/tag08/alleratenT8.mat) und [[alleratenT8.mat]](http://www.compneuro.uni-oldenburg.de/download/kurs08/tag08/alleratenT8.mat) sind die entsprechenden Ergebnisse gespeichert.

* Plotten Sie *alleraten* gegen *allereize* als Einzelpunkte zusammen mit der bereits erstellten linearen Näherung für den in T8A3) bearbeiteten Teil der Daten.
* Erstellen Sie eine neue lineare Näherung für alle Datenpunkte.
* Wird die Näherung besser, wenn man ein Polynom zweiten Gerades verwendet?
* Wie sieht sie für Polynome höheren Grades aus? Verwenden Sie zur Abschätzung der Güte der Näherung
	+ sowohl die mittlere quadratische Abweichung der Messpunkte von der geschätzten Kurve
	+ als auch *S.normr*
* Variieren Sie den Grad des Polynoms systematisch und protokollieren Sie dabei die beiden resultierenden Fehler, um abzuschätzen, wie sich die Anpassung mit wachsender Koeffizientenzahl verbessert.
* Wie viele Koeffizienten werden gebraucht, um die Daten mit *polyfit*zufriedenstellend anzunähern?
	+ Wann wird die Schätzung nicht mehr deutlich besser, wenn man zusätzliche Koeffizienten hinzunimmt?
	+ Wird die Schätzung irgendwann wieder schlechter?
	+ Führen beide Fehlermaße (mittlere quadratisch Abweichung und *S.normr*) zum gleichen Ergebnis?

T8A8) Berechnen Sie für jeden der Stromwerte, mit dem Messungen durchgeführt wurden, Mittelwert und Standardabweichung und plotten diesen mit *errorbar* in eine Abbildung. (Hinweis: für diese Aufgabe ist es günstiger, aus dem langen Vektor der Messwerte eine Matrix zu machen. Das geht mit *reshape*.)

\*T8A9) Mathematisch Interessierte könnten sich unter dem Stichwort "least squares fitting" ansehen, wie die Datenanpassung in Matlab vorgenommen wird und welche Möglichkeiten die Curve Fitting Toolbox noch bietet.

B) VARIATION VON MODELLPARAMETERN

Das Ziel vieler Experimente ist es, eine Abhängigkeit zwischen Parameterwerten und einer Messgröße zu finden. Man versucht, diese Abhängigkeit mathematisch durch eine Formel zu beschreiben. Dadurch kann man für gegebene Parameterwerte vorhersagen, welche Messwerte man erwartet. Dieses Vorgehen bezeichnet man auch als Simulation. Im Experiment kann dann überprüft werden, ob die Vorhersagen den tatsächlichen Messungen entsprechen. Wenn dies nicht der Fall ist, muss das Modell (also die Formel zur Beschreibung der Abhängigkeit) entsprechend angepasst werden.

Hier werden wir als Beispiel für ein Modell die Sigmoidfunktion verwenden. Sigmodidfunktionen spielen als Sättigungsfunktionen in der Biologie eine wichtige Rolle, da es kaum Beispiele für unbegrenzt lineare Abhängigkeiten gibt. Sigmoidfunktionen kann man mit der Formel

*y=a./(1+(exp(b\*(c-x))))*

darstellen.

**Aufgaben:**

T8B1) Tatsächlich handelt es sich bei den gerade verwendeten Daten nicht um echte Messungen, sondern um mit Rauschen überlagerte Punkte auf einer Sigmoidfunktion:

* Erzeugen Sie einen Vektor *x*, der von -10 bis 10 in kleinen Schritten läuft.
* Belegen Sie die Parameter der Sigmoidformel mit den Werten *a=120; b=1; c=2*. Berechnen Sie y entsprechend der Sigmoidformel.
* Plotten Sie die Sigmoidfunktion zusammen mit *alleraten* und Ihrem besten Fit.
* Wie ähnlich sind sich die beiden Linien und die Messpunkte?
* \*) Mit welcher Standardabweichung habe ich die Datenpunkte erzeugt?

 T8B2) Es lohnt sich, die Sigmoidfunktion genauer kennenzulernen. Schreiben Sie ein Programm:

* die drei Parameter der Sigmoidfunktion sollen systematisch variiert werden.
* stellen Sie die Auswirkung jedes der drei Parameter in einem eigenen Grafikfenster als Kurvenscharen dar.
* Überlegen Sie: Was bewirken die drei Parameter?
* (Tipp: gehen Sie von den Werten *a=1, b=1, c=1* aus, um die Effekte gut zu sehen.)

C) SIMULATION VON ZEITREIHEN

Biologische Prozesse sind fast immer Abläufe in der Zeit, bei denen der augenblickliche Zustand von der Vergangenheit abhängt. Außerdem spielt normalerweise Zufall eine große Rolle. Leider haben wir in diesem Kurs keine Zeit, ausführlich auf diese Thematik einzugehen. (Im ICBM werden hierzu spezielle Kurse angeboten.) Deshalb hier nur ein Beispiel für eine Simulation eines solchen zeitabhängigen Zufallsprozesses:

T8C1) Fünf Ratten leben bei dauerhafter Beleuchtung in einem Käfig. Bei unserer Beobachtung interessieren wir uns im Moment nur für die Frage, wie viele der Ratten zu einem gegebenen Zeitpunkt schlafen und wie viele wach sind.

Schreiben Sie eine Simulation des schlaf-wach-Verhaltens der Ratten, in der sie alle 10 Minuten bestimmen, wie viele Ratten wach sind, basierend auf folgenden Regeln:

* Ausgangszustand: 3 Ratten sind wach, 2 schlafen
* Wenn zu einem gegebenen Zeitpunkt nur 0 oder 1 Ratte wach ist, ist die Wahrscheinlichkeit für jede der Ratten 80%, dass sie bei der nächsten Messung schläft.
* Wenn 2 oder mehr Ratten wach sind, ist die Wahrscheinlichkeit für jede Ratte nur 30%, dass sie bei der nächsten Messung schläft (denn die Tiere stören sich gegenseitig.)
* Lassen Sie Ihre Simulation über einen langen Zeitraum laufen und protokollieren Sie die Anzahl der wachen Ratten.
* Wie groß ist der Prozentsatz der Zeit, den die Tiere schlafend verbringen?

T8C2) Jetzt betrachten wir Ratten in Einzelkäfigen: 25 Ratten sind in Einzelkäfigen untergebracht, die in fünf Reihen nebeneinander stehen, so dass sich insgesamt eine 5x5 Matrix aus Käfigen ergibt.

* Für jede der Ratten wird zufällig entschieden, ob sie schläft oder nicht. Die Wahrscheinlichkeit für schlafen beträgt 40%.
* Simulieren Sie dieses zufällige Schlafverhalten in Zeitschritten von 10 Minuten für eine Woche.
* Erzeugen Sie einen "Film" in dem nacheinander für jede Ihrer Messungen graphisch dargestellt ist, in welchen Käfigen die Ratten schlafen und in welchen sie wach sind (schlafend und wach werden durch zwei verschiedene Farben dargestellt).

T8C3) Jetzt werden die Regeln etwas interessanter:

* Von den 25 Ratten stören sich jeweils nur diejenigen in Nachbarkäfigen beim schlafen.
* Wenn alle Nachbarn schlafen, ist die Wahrscheinlichkeit für jede Ratte 80%, dass sie bei der nächsten Messung schläft.
* Wenn ein Nachbar wach ist, sinkt die Wahrscheinlichkeit auf 40%.
* Wenn zwei Nachbarn wach sind, sinkt die Wahrscheinlichkeit auf 30%.
* Wenn mehr als zwei Nachbarn wach sind, schläft die Ratte im nächsten Zeitschritt nur mit 20% Wahrscheinlichkeit.
* Lassen Sie Ihre Simulation über einen langen Zeitraum laufen und sehen sich die wechselnden Muster als "Film" an.
* Protokollieren Sie dabei, welche der Ratten wann wach ist.
* Wie groß ist der Prozentsatz der Zeit, den die einzelnen Tiere schlafend verbringen?
* Gibt es Unterschiede zwischen den Tieren? Sind diese signifikant?

D) HAUSAUFGABEN

T8H1) Sie impfen 20 Petrischalen jeweils mit 100 Einzellern und zählen 10 Stunden lang 10 mal pro Stunde,  wie viele Einzeller sich in der Petrischale befinden. Dadurch erhalten Sie folgende Matrix [[populationen.mat]](http://www.compneuro.uni-oldenburg.de/download/kurs06/populationen.mat).

* Schauen Sie sich die Messpunkte grafisch an.
* Ermitteln Sie das Polynom, das das Wachstum der Population am besten beschreibt.

\*T8H2) In den Neurowissenschaften ist es eine verbreitete Herangehensweise, die Zeitpunkte des Auftretens einzelner Aktionspotentiale als unwichtig anzusehen und nur die Spikerate (also Anzahl aufgetretener Aktionspotentiale in einem gegebenen Zeitfenster) zu betrachten. Wenn dann das Auftreten einzelner Spikes simuliert werden soll, wird dies mit einem Zufallsprozess getan, dem sogenannten Poissonprozess. Ein Poissonprozess hat die Eigenschaft, dass zu jedem Zeitschritt unabhängig von allen anderen Zeitschritten entschieden wird, ob ein Ereignis stattfindet oder nicht. Wenn z.B. ein Neuron mit 10 Spikes/s feuert und eine Sampelrate von 1000Hz benutzt wird, ist die Wahrscheinlichkeit für das Auftreten eines Spikes in jedem Zeitschritt 1%.

a) Schreiben Sie eine Funktion, die als Eingabewerte eine Spikerate, die Samplerate  und die Länge der gewünschten simulierten Antwortspur bekommt und einen Vektor mit  der simulierten Antworten des Neurons zurückgibt.

b) Erweitern Sie Ihre Funktion so, dass sie nicht einen festen Wert für die Spikerate als Eingabe bekommt, sondern einen Vektor, der die Spikerate für jeden Zeitpunkt der zu simulierenden Antwort angibt (die Länge braucht man dann nicht mehr zu übergeben, sie ergibt sich aus der Länge des Vektors der Spikeraten).

\*T8H3) Der Poissonprozess ist das einfachste denkbare Neuronenmodell und kann viele Eigenschaften biologischer Nervenzellen nicht erklären. Ein etwas realistischeres (aber immer noch sehr einfaches)  Modell für Aktionspotentialantworten eines Neurons ist das integrate-and-fire Neuronenmodell. Es handelt sich um ein Differentialgleichungsmodell, das die zeitliche Änderung der Membranspannung beschreibt, in Kombination mit einer Schwelle, die über die Zeitpunkte des Auslösens von Aktionspotentiallen entscheidet.

Das Demoprogramm [[intfire.m]](http://www.compneuro.uni-oldenburg.de/download/Matlab_Einfuehrung/2011/downloads/intfire.m) zeigt, wie ein integrate-and-fire Neuronenmodell auf einen externen Eingangsstrom reagiert.

a) Wie reagiert das Modellneuron auf andere injizierte Stromverläufe?

* Probieren Sie aus, den Puls stärker oder schwächer zu machen.
* Machen Sie den Puls länger oder kürzer.
* Was ist der kürzeste Puls, bei dem ein spike ausgelöst wird?
* Wie reagiert das Modell auf einen Sinusförmigen Reiz?

 b) Was bewirken die Parameter?:

* Welche Auswirkungen hat die Zeitkonstante?
* Welche Auswirkungen hat die Feuerschwelle?
* Was passiert, wenn man von einem anderen Membranpotentialwert startet?

c) Wie wirkt sich Rauschen auf die Antworten aus?

* Erweitern Sie  das Modell, indem Sie bei der Integration in jedem Zeitschritt *tau\_bin\*randn* hinzuaddieren. Dieser Schritt soll simulieren, dass Nervenzellen auf gleiche Reizung nicht jedes mal genau gleich antworten.
* Lassen Sie die Simulation mehrfach laufen. Wie verändern sich die Ergebnisse?
* Führen Sie einen neuen Parameter ein, der die Größe des Rauschens skaliert und probieren Sie seinen Effekt aus.
* Modifizieren Sie das erweiterte integrate-und-fire Skript zu einer Funktion, die die Anzahl erzeugter Spikes zurückgibt aber keine graphische Ausgabe hat.
* Schreiben Sie ein skript, das die neue integrate-and-fire Funktion 100 mal aufruft und ein Histogramm der vorkommenden Spikeanzahlen plottet.

\*\*T8H4) Schreiben Sie ein Programm, das für eine gegebene Menge an Messwerten eine Sigmoidfunktion bestimmt, die den mittleren quadratischen Abstand zu den Messwerten minimiert.

Testen Sie diese Funktion, indem Sie künstliche Daten herstellen, bei denen eine Sigmoidfunktion mit  Rauschen überlagert wird.

T8H5) Bitte gehen Sie nochmal Ihre Notizen aus dem Kurs durch, um morgen noch einmal Fragen stellen zu können!

Außerdem möchte ich Sie darum bitten, die zusätzlichen Texte "[Konzepte](http://www.compneuro.uni-oldenburg.de/46800.html)", "[Programmierung](http://www.compneuro.uni-oldenburg.de/46803.html)" und "[Befehle](http://www.compneuro.uni-oldenburg.de/46799.html)" zu lesen und sich auch dafür Fragen zu überlegen. (Auch wenn Sie das vielleicht vorher schon getan haben sind sie wahrscheinlich jetzt besser verständlich!)

Falls Sie Hinweise zum Skript, den Texten und Aufgaben haben (was unverständlich, missverständlich, falsch oder unvollständig war) wäre ich Ihnen für eine entsprechende Liste sehr dankbar!!!

Und bitte denken Sie an das Ausfüllen des [Rückmeldungsbogens](http://www.compneuro.uni-oldenburg.de/download/Matlab_Einfuehrung/2013/Tag9/Fragebogen_Matlab13.pdf)!