Modellvergleiche

Raphael Hartmann

SUMMER SCHOOL KOGNITIVE MODELLIERUNG 2022

Übersicht

- Anpassungsgüte:
  - χ^2 test
  - G test
- Likelihood Funktion (Repetition)
- Modellvergleich:
  - LR test
  - AIC
  - BIC

χ^2 test

Definition

- Seien X Häufigkeitsdaten (diskret) mit N Beobachtungen und M ein Modell, welches Vorhersagen über die Verteilung von Häufigkeitsdaten in K Kategorien macht. Nach diesem Modell gibt es für jede Kategorie k eine Wahrscheinlichkeit p_k, dass eine zufällige Beobachtung in diese Kategorie fällt. Die Nullhypothese H_0 besagt also, dass die beobachteten Häufigkeiten in den Kategorien x_k den erwarteten Häufigkeiten $m_i = N \cdot p_k$ entspricht. Die Teststatistik

$$X_M^2 = \sum_k^K \frac{(x_k - m_k)^2}{m_k}$$

ist χ^2 verteilt mit $K - 1$ Freiheitsgraden für falls die K Kategorien unabhängig sind und N groß genug. Für ein Modell mit bestimmter Verteilung (Normalverteilung in lin. Reg., Bernoulliverteilung in logist. Reg., Poissonverteilung in Poisson Reg., etc) müssen wir noch die Anzahl an Parametern der Verteilung ($[\mu, \sigma]$, $[\theta]$, $[\lambda]$) abziehen: $K - 1 - m$.

R Funktion

- Der χ^2 test kann wie folgt berechnet werden

```
> chisq.test(x, p)
```

- wobei x ein Vektor mit den K Häufigkeiten ist und p ein Vektor mit den vom Modell abgeleiteten K Wahrscheinlichkeiten ist

Stärken und Schwächen

- Stärken
  - Testbasiert
- Schwächen
  - Testet nur die Anpassungsgüte eines Modells
    - Berücksichtigt keine Sparsamkeit
  - Nur für diskrete (oder diskretisierte) Daten geeignet
  - Ist bei kleinen Zellhäufigkeiten ($x_i < 5$) nicht robust

Kleine Aufgabe

- Berechnen Sie den χ^2 test in folgender Situation:
  - Beobachtete Werte $x = (89, 37, 30, 28, 2)$ und modellbasierte W'keiten $p = (.4, .2, .2, .15, .05)$

- Berechnen Sie die Teststatistik von Hand in R mit der Formel von oben.
  - Nicht vergessen: $m_i = N \cdot p_i$
  - Sie sollten die gleiche Lösung erhalten wie mit der `chisq.test()` Funktion

G test

Auch G^2 test genannt

Definition

- Seien X Häufigkeitsdaten (diskret) mit N Beobachtungen und M ein Modell, welches Vorhersagen über die Verteilung von Häufigkeitsdaten in K Kategorien macht. Nach diesem Modell gibt es für jede Kategorie k eine Wahrscheinlichkeit p_k, dass eine zufällige Beobachtung in diese Kategorie fällt. Die Nullhypothese H_0 besagt also, dass die beobachteten Häufigkeiten in den Kategorien x_k den erwarteten Häufigkeiten $m_i = N \cdot p_k$ entspricht. Die Teststatistik

$$G_M = 2 \cdot \sum_k^K x_k \cdot \ln(x_k/m_k)$$

ist χ^2 verteilt mit $K - 1$ Freiheitsgraden. Für ein Modell mit bestimmter Verteilung (Normalverteilung in lin. Reg., Bernoulliverteilung in logist. Reg., Poissonverteilung in Poisson Reg., etc) müssen wir noch die Anzahl an Parametern der Verteilung ($[\mu, \sigma]$, $[\theta]$, $[\lambda]$) abziehen: $K - 1 - m$.

R Funktion

- Der G test kann wie folgt berechnet werden

```
> library(ARM)
> g.test(x, p)
```

- wobei x ein Vektor mit den K Häufigkeiten ist und p ein Vektor mit den vom Modell abgeleiteten K Wahrscheinlichkeiten ist

Stärken und Schwächen

- Stärken
  - Testbasiert
  - Robuster gegen kleine Zellhäufigkeiten
- Schwächen
  - Testet nur die Anpassungsgüte eines Modells
    - Berücksichtigt keine Sparsamkeit
  - Nur für diskrete (oder diskretisierte) Daten geeignet

Kleine Aufgabe

- Berechnen Sie den G test in folgender Situation:
  - Beobachtete Werte $x = (89,37,30,28,2)$ und modellbasierte W‘keiten $p = (.4, .2, .2, .15, .05)$
  - Installieren Sie hierfür erst mal das R-Paket ARM mit `install.packages("AMR")`
  - Laden Sie danach das Paket ARM mit dem `library()` Befehl

- Berechnen Sie die Teststatistik von Hand in R mit der Formel von oben.
  - Nicht vergessen: $m_i = N \cdot p_i$
  - Sie sollten die gleiche Lösung erhalten wie mit der `g.test()` Funktion

Likelihood Funktion

Repetition

Rückblick

- Die Likelihoodfunktion von mehreren Daten $x = (x_1, \dots, x_N)$ kann folgendermaen dargestellt werden:

$$L(\eta|\, x) = \prod_{i=1}^{N} L(\eta|\, x_i)$$

- Für die Parameterschätzung ist aber die log-Likelihood interessanter, da diese viele Rechnungne vereinfacht:

$$l(\eta|\, x) = \sum_{i=1}^{N} l(\eta|\, x_i)$$

- Im Folgenden kürzen wir die Likelihoodfunktion mit L und die log-Likelihoodfunktion mit l ab

Rückblick

- Maximum Likelihood:
  - Der Wert der Likelihoodfunktion (oder log-Likelihoodfunktion), der am größten ist.
  - Formal ausgedrückt:

$$\hat{L}(\eta|x) = \max_{\eta \in \mathbb{R}} L(\eta|x)$$

- Im Folgenden kürzen wir die Maximum Likelihoodfunktion mit $\hat{L}$und die Maximum log-Likelihoodfunktion mit $\hat{l}$ ab

Devianzmaße

- Die Devianz ist ein Maß der Anpassungsgüte (*goodness-of-fit*)
  - Es wird genutzt für Hypothesentestung und Modellvergleiche
- Es gibt unterschiedliche Devianzmaße, je nachdem, was man erreichen möchte:
  - Devianz $= -2\left(\ln(\hat{L}_{M_0}) - \ln(\hat{L}_{M_1})\right)$:
    - Vergleich des interessierenden Modelles M_1 mit einem Null-Modell M_0 (z. B. für GLMs ein Modell ohne Prädiktoren)
  - Modell-Devianz $= -2\ln(\hat{L}_{M_1})$:
    - Vergleich des interessierenden Modelles M_1 mit einem hypothetisch perfekten Modell
  - Relative Devianz $= -2\left(\ln(\hat{L}_{M_1}) - \ln(\hat{L}_{M_2})\right)$:
    - Vergleich des interessierenden Modelles M_1 mit einem anderen explizit formulierten Modell M_2

R Funktion für GLMs

- Die Modell Devianz kann wie folgt berechnet werden

```
> -2*logLik(model1)
```

- wobei `model1` das R Objekt für das Modell M_1 ist
- Die Devianz oder die Relative Devianz lässt sich berechnen mit

```
> -2*(logLik(model0) - logLik(model1))
> -2*(logLik(model1) - -2*logLik(model2))
```

Aufgabe

- Rechnen Sie eine lineare Regression (mit der `lm()` Funktion in R und den Daten `df_modelselection.RData`)
  - mit der Gleichung $Y_i = \beta_0 + \epsilon_i$ (Nullmodell: `lm0`):
    ```
    > lm0 <- lm(formula = …, data = df)
    ```
  - mit der Gleichung $y = Y_i = \beta_0 + \beta_1 x_1 + \epsilon_i$ (Modell mit einem Prädiktor: `lm1`)
    - `lm1 <- lm(formula = …, data = df)`
  - mit der Gleichung $y = Y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i$ (Modell mit zwei Prädiktoren: `lm2`)
    - `lm2 <- lm(formula = …, data = df)`
- Rechnen Sie für `lm1` nun die Modell Devianz, die Devianz (in Vergleich zu `lm0`) und die relative Devianz (im Vergleich zu `lm2`)
- Nutzen Sie die Funktionen aus Annes Funktionen (im Skript für Sie vorbereitet), um selbst die Modell Devianz für `lm1` zu berechnen

Vorteil von Devianzmaßen

- Die Devianzmaße sind χ^2 verteilt mit Freiheitsgraden (df) abhängig von der Anzahl Parameter m im Modell M_i (also m_{M_i}):
  - Devianz $\sim \chi^2(df = m_{M_1} - m_{M_0})$
  - Modell-Devianz $\sim \chi^2(df = m_{M_{voll}} - m_{M_0})$
  - Relative Devianz $\sim \chi^2(df = m_{M_2} - m_{M_1})$

Likelihood ratio test (LRT)

Definition

- Seien M_0 und M_1 zwei genestete Modelle (M_0 ist Spezialfall von M_1), wobei M_0 k Parameter hat und M_1 k + m Parameter hat. $\boldsymbol{\theta} = (\theta_{k+1}, \dots, \theta_{k+m})$ sind also die Parameter, die M_1 mehr hat als M_0 und H_0 besagt, dass $\boldsymbol{\theta} = \boldsymbol{c}$ (ein Vektor aus Konstanten), dann ist die LR **Teststatistik** definiert als

$$\lambda_{LR} = -2 \ln\left(\frac{\hat{L}_{M_0}}{\hat{L}_{M_1}}\right) = -2\left(\ln(\hat{L}_{M_0}) - \ln(\hat{L}_{M_1})\right) = -2(\hat{l}_{M_o} - \hat{l}_{M_1})$$

- λ_{LR} (=Devianz) ist asymptotisch χ^2_m verteilt, falls H_0 gilt
- Das Gleiche geht auch mit anderen Devianzmaßen (siehe 2 Folien zuvor)
- Es gibt einige Tests, die äquivalent sind zum LRT, die aber nicht so teststark sind:
  - Z-test, F-test, G-test, Pearson‘s χ^2 test

R Funktion für GLMs

- Der LRT kann wie folgt berechnet werden
  - Entweder mit dem R Paket lmtest und der Funktion:

    ```
    > lrtest(model0, model1)
    ```

  - Oder mit der Funktion aus dem Standard:

    ```
    > anova(model0, model1, test="LRT")
    ```

- wobei `model0` das R Objekt für das Modell M_0 und `model1` das R Objekt für das Modell M_1 ist.
- Zudem berechnet `anova()` eine analoge Version des Likelihood Ratio tests über Quadratsummen, also nicht genau das, was wir wollen. Die p-Werte sollten aber übereinstimmen.

Stärken und Schwächen

- Stärken
  - Testbasiert
- Schwächen
  - Modelle müssen genestet sein

Aufgabe

- Berechnen Sie die **Teststatistik** des LRT für die beiden linearen Modelle M_0 $(Y_i = \beta_0 + \epsilon_i)$ und M_1 $(Y_i = \beta_0 + \beta_1 X + \epsilon_i)$
  - einerseits mittels der Funktion `lrtest()` und
  - andererseits indem Sie die Formel und Annes Funktionen (im Skript vorbereitet für Sie) nutzen in R. (Die Modelldevianz für das größere Modell haben wir ja schon berechnet, nämlich M_Devianz1. Es fehlt nur noch M_Devianz0)

Akaike information criterion (AIC)

Definition

- Sei k die Anzahl geschätzter Parameter im Modell M (bspw. \beta-Gewichte und Residualvarianz im linearen Modell) und sei $\hat{L}_M$ die Maximum Likelihood des Modells M, dann ist der AIC folgendermaßen definiert

$$AIC_M = 2k - 2\ln(\hat{L}_M) = 2k - 2\hat{l}_M$$

- Der AIC hat also zwei Komponenten:
  - Devianz (Anpassungsgüte): $-2\ln(\hat{L}_M)$
  - Strafterm für die Anzahl Parameter (Sparsamkeit): $2k$
    - Strafe für Overfitting: Hinzufügen von Parametern führt fast immer zu besserer Anpassungsgüte

Interpretation

- Je kleiner der Wert (bzw. je negativer), desto besser
- Für zwei Modelle M_1 und M_2 mit den jeweiligen AIC Werten AIC_1 und AIC_2 gilt
  - Falls $AIC_1 < AIC_2$: Modell M_1 ist besser als Modell M_2
  - Falls $AIC_1 > AIC_2$: Modell M_1 ist schlechter als Modell M_2
- Ab wann ist ein Unterschied zwischen zwei AIC Werten bedeutsam?
  - Gängige Praxis ist der Wert 4: Also für $|AIC_1 - AIC_2| > 4$
  - Falls $|AIC_1 - AIC_2| < 4$, so wählt man das sparsamere Modell oder berücksichtigt beide

R Funktion für GLMs

- Der AIC eines Modelles kann wie folgt berechnet werden

```
> AIC(model)
```

- wobei `model` das R Objekt für das gefittete Modell ist

Stärken und Schwächen

- Stärken
  - Berücksichtigung von Sparsamkeit
  - Modelle müssen nicht genestet sein
- Schwächen
  - Für kleines N werden Modelle präferiert, die zu viele Parameter haben
    - Es gibt eine Korrektur hierfür
  - Es berücksichtigt nicht die Stichprobengröße

Kleine Aufgabe

- Berechnen Sie den AIC für ein lineares Modell ($Y_i = \beta_0 + \beta_1 X + \epsilon_i$)
  - einerseits mittels der Funktion `AIC()` und
  - andererseits von Hand bzw. indem Sie die Formel nutzen in R. (Die Modell Devianz haben wir ja schon berechnet)

Bayesian information criterion (BIC)

Definition

- Sei k die Anzahl geschätzter Parameter im Modell M (bspw. \beta-Gewichte und Residualvarianz im linearen Modell), N die Anzahl Datenpunkte (bspw. Stichprobengröße) und sei $\hat{L}_M$ die Maximum Likelihood des Modells M, dann ist der BIC folgendermaßen definiert

$$BIC_M = k\ln(N) - 2\ln\left(\hat{L}_M\right) = k\ln(N) - 2\hat{l}_M$$

- Der BIC hat also zwei Komponenten:
  - Devianz (Anpassungsgüte): $-2\ln\left(\hat{L}_M\right)$
  - Strafterm für die Anzahl Parameter (Sparsamkeit): $k\ln(N)$
    - Strafe für Overfitting: Hinzufügen von Parametern führt fast immer zu besserer Anpassungsgüte
    - Der Strafterm ist härter als der vom AIC für $N \geq 8$.

Interpretation

- Je kleiner der Wert (bzw. je negativer), desto besser
- Für zwei Modelle M_1 und M_2 mit den jeweiligen BIC Werten BIC_1 und BIC_2 gilt
  - Falls $BIC_1 < BIC_2$: Modell M_1 ist besser als Modell M_2
  - Falls $BIC_1 > BIC_2$: Modell M_1 ist schlechter als Modell M_2
- Ab wann ist ein Unterschied zwischen zwei BIC Werten bedeutsam?
  - Gängige Praxis ist der Wert 4: Also für $|BIC_1 - BIC_2| > 4$
  - Falls $|BIC_1 - BIC_2| < 4$, so wählt man das sparsamere Modell oder berücksichtigt beide

R Funktion für GLMs

- Der BIC eines Modelles kann wie folgt berechnet werden

```
> BIC(model)
```

- wobei `model` das R Objekt für das gefittete Modell ist

Stärken und Schwächen

- Stärken
  - Berücksichtigung von Sparsamkeit
  - Modelle müssen nicht genestet sein
  - Berücksichtigung von Stichprobengröße
- Schwächen
  - N muss viel größer als k sein.

Kleine Aufgabe

- Berechnen Sie den BIC für ein lineares Modell ($Y_i = \beta_0 + \beta_1 X + \epsilon_i$)
  - einerseits mittels der Funktion `BIC()` und
  - andererseits von Hand bzw. indem Sie die Formel nutzen in. (Die Modell Devianz haben wir ja schon berechnet)

Zusammenfassung

Kennwerte für Modellgüte und -vergleiche

	χ^2 test	G test	LR test	AIC	BIC
Testbasiert	**ja**	**ja**	**ja**	nein	nein
Daten	diskret	diskret	**alles**	**alles**	**alles**
Modellvergleich	nein	nein	**ja**	**ja**	**ja**
Sparsamkeit	nein	nein	**(ja)**	**ja**	**ja**
Nestung	-	-	ja	**nein**	**nein**
R Funktion	`chisq.test(x, p)`	`ARM::g.test(x, p)`	`anova(m0, m1, test="LRT")`	`AIC(m1)`	`BIC(m1)`

Übungen

Aufgabe 1

- Nutzten Sie den Datensatz `alcohol.RData`, um logistische Regressionen zu rechnen mit der R Funktion `glm(formula, data, family = binomial(link = "logit"))`.
  - Speichern Sie folgende Modelle als R Objekte ab
    - Modell 0: Nullmodell (keine Prädiktoren) → bspw. speichern als `glm0`
    - Modell 1: Modell mit Prädiktor Stress → bspw. speichern als `glm1`
    - Modell 2: Modell mit Prädiktoren Stress und Support → bspw. speichern als `glm2`

Aufgabe 2

- Rechnen Sie nun für alle Modelle die **Modell Devianz.**
- Rechnen Sie für Modell 1 zusätzlich die Devianz und relative Devianz

Aufgabe 3

- Führen Sie einen **Likelihood Ratio test** durch, indem Sie alle Modelle miteinander vergleichen.

- Berechnen Sie zusätzlich für jedes Modell den **AIC** und **BIC**
  - Kommen beide zum gleichen Schluss
  - Vergleichen Sie noch mit dem Likelihood Ratio test: Kommt dieser zum gleichen Schluss wie AIC und/oder BIC?

Aufgabe 4

- Berechnen Sie die Modell Devianz mit dem etwas angepassten **Optimierungsalgorythmus** (über Devianzen) von Anne für das Modell 2 (beide Prädiktoren). **Den Code dazu finden Sie auf der nächsten Folie. Einige Stellen (rot markiert) müssen Sie noch richtig ergänzen.**

- Vergleichen Sie den Wert zu dem Modell Devianz Wert, den Sie über `glm()` erhalten haben.

- Berechnen Sie noch den **AIC** und **BIC** mittels der gerade ermittelten Modell Devianz

Aufgabe 4 Forts.

```
# Achtung: wir haben nun zwei Praediktoren
PredReg2 <- function(para, predictor1, predictor2) {
  b0 <- para[1]
  b1 <- para[2]
  b2 <- para[3]
  Xbeta <- # hier kommt der lineare Praediktor mit para und predictor1
und 2
  theta <- # hier kommt die Formel, um theta zu berechnen (siehe
vorletzte Sitzung)
  return (theta)
}

Deviance2 <- function(para, data) {
  criterium <- data[,1]
  predictor1 <- # hier kommt der Praediktor (Stress) ueber data[,...]
  predictor2 <- # hier kommt der Praediktor (Support) ueber data[,...]
  PredictedTheta <- PredReg2(para, predictor1, predictor2)
  likelihood <- # hier kommt die PMF (dbinom) mit den entsprechenden
Argumenten (nicht vergessen: size = 1)
  deviance <- -2*sum(log(likelihood))
  return(deviance)
}
```

```
minvalue <- 10^10

# Achtung, wir brauchen 3 Parameter -> startpar hat 3 Elemente
for(run in 1:5) {
  startpar <- c(runif(1, -5, 5),runif(1, -5, 5), runif(1, -5,5))
  fit <- optim(par = startpar, fn = Deviance2, data = alcohol)

  if(fit$value < minvalue) {
    bestfitDeviance2 <- fit
    minvalue <- bestfitDeviance2$value
  }
}
(M_Devianz2 <- bestfitDeviance2$value)
```