Python-Beleg zum Modul Biostatistik

PD Dr. Steffen Löck
Prof. Dr. Wolfgang Enghardt

Sommersemester 2015
Abgabe bis 12.10.2015

Die folgende Aufgabe ist Bestandteil der Prüfung zum Modul Biostatistik (50 %). Bitte erstellen Sie das entsprechende Programm sowie die geforderte Dokumentation, und geben Sie beides in elektronischer Form (per E-Mail) ab.

Aufgabenstellung

Gegeben sei die Excel-Tabelle `Beleg_Biostatistik_2015_Daten.xls`, die (veränderte) Daten von Patienten mit lokal fortgeschrittenem Kopf-Hals-Tumor, welche am UKD innerhalb der FMISO Studie primär bestrahlt wurden, enthält. Die Daten bestehen aus einer Experimentalkohorte und einer Validierungskohorte. Ziel dieser prospektiven Studie war es, mit der ersten Kohorte den prognostischen Wert einer Hypoxiebestimmung mittels FMISO-PET im Verlauf der Strahlentherapie (0., 1., 2. und 5. Woche) zu bestimmen (Zips et al. Radiother Oncol, 105 (2012) 21-28) und das Ergebnis mit der zweiten Kohorte zu validieren.

Schreiben Sie ein Python-Programm, mit dem man unter Verwendung der Cox-Regression die Korrelation zwischen rezidivfreiem Überleben und klinischen sowie Hypoxie-Kovariablen bestimmen kann.

Arbeiten Sie dazu zunächst die Erklärungen zur Cox-Regression am Ende dieses Dokuments durch.

Im Detail soll das Programm folgendes leisten:

- Laden Sie die Excel-Datei mit Python, entweder direkt oder nach Umwandlung in eine Textdatei.
- Es soll im folgenden vom Nutzer ausgewählt werden können, ob eine Cox-Regression der ersten Kohorte, der zweiten Kohorte oder der kombinierten Daten durchgeführt werden soll.
- Danach soll der Nutzer die zu berücksichtigenden klinischen- und/oder Hypoxie-Kovariablen wählen können (eine oder mehrere).
- Wenn der Nutzer k Kovariablen gewählt hat, soll ein k-parametriges Cox-Modell an die Daten der vom Nutzer ausgewählten Kohorte gefittet werden. Geben Sie die gefitteten Parameter, deren Standardabweichung, die Hazard-Ratio, deren Konfidenzintervall, Wald-Statistik, p-Wert sowie eine Aussage, ob die betrachtete Kovariable einen signifikanten Einfluss auf des

Überleben hat, auf der Konsole übersichtlich aus. Dabei soll das Signifikanzniveau vom Nutzer eingegeben werden können.

- Zeichnen Sie für univariate Modelle die logarithmierte Überlebensfunktion $S(t)$ mit der Annahme $h_0(t) = 1$ für den unbekannten Baseline-Hazard. Bei einer binären Kovariable zeichnen Sie die Überlebensfunktionen der beiden Gruppen. Bei einer kontinuierlichen Kovariable zeichnen Sie die Überlebensfunktionen für das 0 %, 25 %, 50 %, 75 % und 100 % Perzentil der Kovariable. Achten Sie auf optische Unterscheidbarkeit und verwenden Sie eine Legende. Bei multivariaten Modellen soll nichts gezeichnet werden.

- Geben Sie dem Benutzer nach jeder Berechnung die Möglichkeit Kohorte, zu berücksichtigende Kovariablen und Signifikanzniveau neu zu wählen oder sich für das Programmende zu entscheiden.

Analysieren Sie mit Ihrem Programm die gegebenen Daten und dokumentieren Sie die Ergebnisse: Führen Sie folgende Cox-Regressionen durch und stellen Sie fest, welche Kovariablen unter Verwendung des Wald-Tests einen signifikanten Zusammenhang mit dem rezidivfreien Überleben haben ($\alpha = 5\%$):

1. Testen Sie alle Kovariablen univariat für die kombinierten Daten (erste und zweite Kohorte kombinert).

2. Erstellen Sie alle multivariaten Modelle der kombinierten Daten mit den klinischen Kovariablen, die in 1. zumindest einen statistischen Trend aufweisen, und jeweils einer Hypoxie-Variable. Interpretieren Sie die Ergebnisse. Welcher Zeitpunkt hat die größte Vorhersagekraft?

3. Teilen Sie die Patienten der ersten Kohorte in zwei gleich große Gruppen, indem Sie diese Patienten nach dem Median der Variable TBR_2 trennen. Definieren Sie dazu eine neue Variable (0 oder 1). Führen Sie eine univariate Cox-Regression für die erste Kohorte mit dieser neuen Gruppenvariable durch.

4. Validieren Sie den Cut-off (Median) der ersten Kohorte: Teilen Sie die Patienten der zweiten Kohorte in zwei Gruppen, indem Sie diese Patienten nach dem Median der ersten Kohorte in der Variable TBR_2 (aus 3.) trennen. Definieren Sie dazu eine neue Variable (0 oder 1). Führen Sie eine univariate Cox-Regression für die zweite Kohorte mit dieser neuen Gruppenvariable durch.

5. Interpretieren Sie das Ergebnis aus 3. und 4. und überlegen Sie sich mögliche Ursachen anhand der vorliegenden Daten.

Speichern Sie alle Resultate und binden Sie sie in ein Word- oder pdf-Dokument ein. Dokumentieren Sie die ermittelten Fitparameter, deren Standardabweichungen, die Hazard-Ratios, deren 95%-Konfidenzintervalle, Wald-Statistik sowie p-Werte für alle Regressionen. Diskutieren Sie die möglichen Schlüsse dieser Analyse für die betrachtete Studie.

Achten Sie auf einen sauberen und effizienten Programmierstil; verwenden Sie Blöcke und Einrückungen sowie aussagekräftige Variablennamen. Kommentieren Sie den Quelltext ausführlich, so dass sich auch andere Programmierer in Ihr Programm hineindenken können. Die Cox-Regression sowie der dabei auftretende iterative Prozess zur Bestimmung der optimalen Fitparameter soll selbst programmiert werden (keine externen Python Module verwenden). Arbeiten Sie selbstständig.

Zusatzaufgaben:

1. Oft werden Likelihood-Quotiententests gegenüber Tests basierend auf der Wald-Statistik bevorzugt. Fügen Sie in Ihr Programm die Möglichkeit ein, dass sich der Nutzer vor der Berechnung zwischen dem Wald-Test und dem Likelihood-Quotiententest entscheidet. Geben Sie im Fall der Entscheidung für den Likelihood-Quotiententest statt der Wald-Statistik die benötigte Differenz der Log-Likelihoods (multipliziert mit 2) auf der Konsole aus und berechnen daraus den entsprechenden p-Wert. Diskutieren Sie, wie stark sich die Ergebnisse der beiden Testmethoden unterscheiden.

2. Schreiben Sie Ihr Programm objektorientiert. Erstellen Sie eine Klasse für die numerische Berechnung der Parameter der Cox-Regression und eine für die restliche Funktionalität des Beleges.

Hinweise zur Bewertung

Für Programm und Dokumentation werden 85 % der Punkte vergeben. Weitere 15 % werden für Programmierstil sowie Kommentierung vergeben. Durch Bearbeitung der Zusatzaufgaben können Sie zusätzlich jeweils 5 % erwerben.

Wir wünschen Ihnen viel Erfolg!

Erläuterungen zur Cox-Regression

Die Cox-Regression sagt das zeitabhängige Überleben $S(t)$ von Individuen im Vergleich zu einer Baseline-Gruppe voraus, für welche alle berücksichtigten Kovariablen gleich Null sind. Für jeden Patient ist eine Überlebenszeit t_i, die binäre Information, ob das zu untersuchende Ereignis eingetreten ist oder nicht (Zensierung) δ_i sowie beliebig viele Kovariablen (z.B. Alter, Geschlecht, ...) x_{ij} gegeben. Es gebe N Patienten ($i = 1 \dots N$) sowie M Kovariablen ($j = 1 \dots M$). Die Cox-Regression fittet den konkreten Ansatz für die Hazard-Funktion

$$h(t, \mathbf{x}) = h_0(t) \, \exp\left(\sum_{j=1}^{M} b_j x_j\right) = h_0(t) \cdot \exp(b_1 x_1) \cdot \ldots \cdot \exp(b_M x_M) \qquad (1)$$

an die Überlebensdaten der betrachteten Patienten durch die Optimierung der Fitparameter b_j. Die Kovariablen x_{ij} gehen exponentiell und zeitunabhängig in die Hazard-Funktion ein, welche über

$$h(t) = -\frac{\mathrm{d}}{\mathrm{d}t} \ln(S(t)) \qquad (2)$$

mit der Überlebensfunktion $S(t)$ verknüpft ist. Eine Annahme der Cox-Regression ist es, dass die Hazard-Funktion $h(t, \mathbf{x})$ nur durch den zeitunabhängigen Faktor $\exp(\sum_j b_j x_j)$ von der unbekannten Baseline-Hazard-Funktion $h_0(t)$ abweicht. Diese Bedingung nennt man die proportionale-Hazard-Annahme. Sie soll im Rahmen dieses Beleges nicht geprüft werden. Wird in Gleichung (1) nur eine Kovariable eingebracht ($M = 1$), so nennt man die Analyse univariat. Sind es mehr als eine ($M > 1$), dann handelt es sich um eine multivariate Analyse. Im Folgenden bezeichnen fettgedruckte Kleinbuchstaben Vektoren, z.B. $\mathbf{b} = (b_0, b_1, ..., b_M)$, und fettgedruckte Großbuchstaben Matrizen, z.B. $\mathbf{X} = (x_{ij})$.

Gegeben sei ein konkretes Patientenkollektiv mit Überlebenszeiten t_i, Zensierungen δ_i sowie Kovariablen $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,M})$. Dabei gilt $\delta_i = 1$, wenn zur Zeit t_i ein Ereignis eingetreten ist und $\delta_i = 0$ kennzeichnet eine Zensierung zur Zeit t_i. Ziel ist es, die Fitparameter b_j in Gleichung (1) so zu bestimmen, dass die Hazard-Funktion die Patientendaten bestmöglich reproduziert. Dazu wird die partielle Likelihood-Funktion L verwendet. Sie ist für den Zeitpunkt t_i gegeben als

$$L_i(\mathbf{b}) = \frac{h_0(t) \, \exp\left(\sum_{j=1}^{M} b_j x_{ij}\right)}{\sum_{k:t_k \geq t_i} h_0(t) \, \exp\left(\sum_{j=1}^{M} b_j x_{kj}\right)}, \qquad (3)$$

wobei im Nenner über alle Zeiten größer gleich t_i summiert wird. Die Baseline-Hazard-Funktion kürzt sich heraus. Die gesamte Likelihood-Funktion ergibt sich dann zu

$$L(\mathbf{b}) = \prod_{i=1}^{N} L_i^{\delta_i}. \qquad (4)$$

Der optimale Fit wird erreicht, wenn die Likelihood-Funktion $L(\mathbf{b})$ maximal ist. Zur Vereinfachung geht man durch logarithmieren von L zur logarithmischen Likelihood-Funktion LL über

$$LL(\mathbf{b}) := \ln[L(\mathbf{b})] = \sum_{i=1}^{N} \delta_i \left\{ \sum_{j=1}^{M} b_j x_{ij} - \ln \left[\sum_{k:t_k \geq t_i} \exp \left(\sum_{j=1}^{M} b_j x_{kj} \right) \right] \right\}. \quad (5)$$

Aus dem Produkt in Gleichung (4) ist eine Summe geworden, die sich leichter optimieren lässt.

Zur Bestimmung des optimalen Parameters $\hat{\mathbf{b}}$ muss Gleichung (5) maximiert werden. Dies muss iterativ erfolgen und soll hier mit der Newton-Raphson-Methode umgesetzt werden. Gegeben sei ein Anfangsschätzer für den Parametervektor $\mathbf{b}^{(l=0)}$ sowie folgende Ableitungen und Definitionen

$$g_i := \sum_{k:t_k \geq t_i} \exp \left(\sum_{j=1}^{M} b_j x_{kj} \right), \quad (6)$$

$$h_{ij} := \frac{\partial g_i}{\partial b_j} = \sum_{k:t_k \geq t_i} x_{kj} \exp \left(\sum_{l=1}^{M} b_l x_{kl} \right), \quad (7)$$

$$A_{ijk} := \frac{\partial g_i}{\partial b_j \partial b_k} = \frac{\partial h_{ij}}{\partial b_k} = \sum_{m:t_m \geq t_i} x_{mj} x_{mk} \exp \left(\sum_{l=1}^{M} b_l x_{kl} \right), \quad (8)$$

$$u_j = \frac{\partial LL(\mathbf{b})}{\partial b_j} = \sum_{i=1}^{N} \delta_i \left\{ x_{ij} - \frac{h_{ij}}{g_i} \right\}, \quad (9)$$

$$I_{jk} = \frac{\partial LL(\mathbf{b})}{\partial b_j \partial b_k} = -\sum_{i=1}^{N} \delta_i \left\{ \frac{1}{g_i} \left(A_{ijk} - \frac{h_{ij} h_{ik}}{g_i} \right) \right\}. \quad (10)$$

Damit ergibt sich ein verbesserter Parametervektor als

$$\mathbf{b}^{(l+1)} = \mathbf{b}^{(l)} - {\mathbf{I}^{(l)}}^{-1} \cdot \mathbf{u}^{(l)}. \quad (11)$$

Dieses Schema wird iterativ für $l = 0, 1, 2, \ldots$ wiederholt, bis sich die Log-Likelihood, Gleichung (5), weniger als eine Schwelle (z. B. 10^{-8}) im aktuellen Iterationsschritt verändert. Bitte beachten Sie, dass in Gleichung (11) eine Matrixmultiplikationen durchzuführen ist und die inverse Matrix von $\mathbf{I}$ benötigt wird. Als Ergebnis erhält man den Vektor $\hat{\mathbf{b}}$ dessen Komponenten die Maximum-Likelihood-Schätzungen der Einzelparameter des Regressionsmodells enthalten. Für den Beleg sind die Startwerte $\mathbf{b}^{(0)} = (0, 0, \ldots, 0)$ sinnvoll. Programmieren Sie mindestens zwei Summationsebenen mit Vektoroperationen (ohne Schleifen)!

Die negative inverse Matrix der zweiten Ableitung der Log-Likelihood-Funktion ist die Varianz-Kovarianz-Matrix

$$\mathrm{Cov}(\hat{\mathbf{b}}) = -\mathbf{I}^{-1}. \quad (12)$$

Sie wird im Newton-Raphson-Algorithmus bereits berechnet. Die Wurzeln der Diagonalelemente von $\mathrm{Cov}(\hat{\mathbf{b}})$ sind Schätzer für die Standardabweichungen $s_{\hat{b}_j}$ der Parameter $\hat{\mathbf{b}}$. Für große N sind die Einzelparameter $\hat{b}_j$ jeweils annähernd normalverteilt. Daraus ergeben sich die Konfidenzintervalle der $\hat{b}_j$ zu

$$\hat{b}_j \pm z_{1-\alpha/2}\sqrt{\mathrm{Cov}(\hat{\mathbf{b}})_{jj}}. \tag{13}$$

Die Hazard-Ratio der Einzelparameter ist definiert als $HR_j = \exp(\hat{b}_j)$. Sie gibt an um wieviel das Risiko, ein Ereignis zu haben, steigt, wenn die Kovariable x_j um 1 wächst. Die Konfidenzintervalle der Hazard-Ratio berechnen sich entsprechend aus denen der Fitparameter.

Zum Testen der Nullhypothese, dass ein Regressor $\hat{b}_j$ gleich Null ist, kann die Wald-Statistik verwendet werden, die bei großem N annähernd χ^2-verteilt ist. Bei dichotomen und metrischen Variablen x_j gilt $W_j = (\hat{b}_j/s_{\hat{b}_j})^2$, was einer χ^2-Verteilung mit einem Freiheitsgrad folgt. Dies erlaubt die Berechnung eines p-Wertes für den Parameter $\hat{b}_j$. Alternativ kann ein Likelihood-Quotiententest durchgeführt werden. Dazu wird die Log-Likelihood des gefitteten Modells LL_1 bestimmt, sowie die Log-Likelihood des Modells, was aus dem betrachteten durch Weglassen des zu untersuchenden Parameters entsteht, LL_2. Die Differenz $2(LL_2 - LL_1)$ ist annähernd χ^2-verteilt mit der Anzahl an fehlenden Parametern als Freiheitsgrad (hier 1). Dies erlaubt die Berechnung eines p-Wertes. Die Differenz der Log-Likelihoods entspricht einem Quotienten der Likelihoods, wodurch der Name des Tests, Likelihood-Quotiententest, zustandekommt.