Hierarchische Modelle in BUGS

Ein Beispiel aus der Analyse von Microarray Data

Ulrich Mansmann, Institut für Medizinische Biometrie und Informatik Universität Heidelberg

1. Was ist ein hierarchisches Modell

Biometrische und medizinische Daten sind oft mit einer hierarchsichen Struktur versehen. Diese Struktur spiegelt in manchen Anwendungen Strukturen im vorliegenden realen Problem wieder: In multizentrischen Studien sind Patienten in Kliniken zusammengefaßt. Es gibt aber auch Probleme, bei denen eine hierarchische Form offensichtlicherweise nicht vorliegt, durch deren Einführung das Problem jedoch besser behandelbar wird: Longitudinale Daten lassen sich, wenn angebracht, durch die Einführung von Zufallseffekten einfacher behandeln. Verschiedenen Zeitpunkte werden als unabhängig angesehen, wenn man auf das einzelne Individuum bedingen kann.

Einen guten Überblick über hierarchische Modelle liefert Harvey Goldstein [Goldstein, 1995].

2. Was ist ein Microarray

Ein cDNA Microarray besteht aus tausenden einzelnen Genen, die sehr dicht auf einer kleinen Glasscheibe aufgebracht werden. Mittels diesem Plättchen wird nun versucht, die relative Häufigkeit der aufgetragenen Gene in den DNAs zweier *Typen* (zweier Zelltypen: Normalgewebe/Tumor oder von zwei Stufen der zeitlichen Entwicklung eines Zelltypes) zu bestimmen, indem man die relative Hybridisierung von Proben der beiden *Typen* an den Sequenzen des Microarray quantifiziert. Um dies möglich zu machen, werden die DNA Proben der beiden *Typen* mit zwei verschieden floureszierenden Farbstoffen markiert (Farbstoff Cy5 für eine rotes Floureszieren und Farbstoff Cy3 für ein grünes Floureszieren). Beide DNAs werden gemischt, auf die Platte aufgebracht. Es findet dann eine konkurrierende Hybridisierung statt. Das Ergebnis wird mit einem Laser abgelesen, der jeden Spot des Arrays abtastet. Das Verhältnis der Farbintensitäten an den einzelnen Spots wird als Maß für den Expressionsunterschied eines Genes zwischen beiden *Typen* genommen. Wichtig sind Normierungsmaßnahmen, die vor der Analyse der Rohdaten vorgenommen werden müssen, wie etwa die Korrektur der Intensitäten gegen das Rauschen in der Umgebung eines Genspots. Microarrays werden in der Diagnostik zur Tumorklassifikation erprobt und beim Studium von Tumormarkern eingesetzt. Dies geschieht in der Hoffnung, dadurch zu einer

besseren Tumorklassifikation zu finden. Beispielhafte Arbeiten zu diesem Problem sind [Alizadeh et al., 2000] und [Golub et al., 1999].

3. Das Gamma-Gamma Modell für Genexpression

Netwton et al. beschrieben in ihrem Artikel *On differntial variablilty of expression ratios: Improving statistical inference about gene expression changes from microarray data* [Newton et al., 2000] ein Modell, das die Genexpression mit einer Gammverteilung modelliert.

Von verschiedenen Genen werden Intensitäten der Farben Rot (R) und Grün (G) gemessen. Für das Expressionsverhältnis eines Genes $\rho = \mu_R / \mu_G$ wird in der Regel der naive Schätzer $\hat{\rho}_{naiv} = R/G$ genommen, wobei R (bzw. G) die mit Meßfehlern behaftete Messung der wahren Expressionsintensität μ_R (bzw. μ_R) ist. Newton et al. kritisieren diesen Schätzer und bieten einen korrigierten *empirical Bayes* Schätzer der folgenden Form an:

$$\hat{\rho}_{EB} = \frac{R + \upsilon}{G + \upsilon}$$

wobei ν ein sich aus der Prozedur ergebender Shrinkage-Faktor ist.

3.1. Newton's Gamma-Gamma-Modell (GG)

Newton's Modell liegt die folgende Überlegung zu Grunde: **In der ersten Stufe des hierarchischen Modells** werden folgende Annahmen an die Verteilung der gemessenen Intensitäten R und G für Gen X gemacht: Die Messung der Intensität R ist mit Skalierungsparameter θ_{RX} (= α_1/μ_{RX}) und Formparameter α_1 gamma-verteilt. Dies gilt analog für die gemessene Intensität von Grün (G) an Gene X, die mit dem Skalierungsparameter θ_{GX} (= α_1 / μ_{GX}) und dem gleichen Formparameter α_1 gamma-verteilt ist. Formal ist die Dichte der Verteilung von R somit gegeben als:

$$p(r \mid \theta_{RX}, \alpha_1) = \theta_{RX}^{\alpha_1} r^{\alpha_1 - 1} \exp[-r \cdot \theta_{RX}] / \Gamma(\alpha_1)$$

Die beiden Skalierungsparameter für die Intensitäten werden als vom spezifischen Gen abhängig angenommen. Der Formparameter α_1 ist für alle betrachteten Gene gleich. Somit gilt für das interessierende Expressionsverhältnis von Gen X:

$$\rho_X = \frac{\mu_R}{\mu_G} = \frac{\theta_G}{\theta_R}.$$

In der zweiten Stufe des hierarchischen Modells werden die Skalierungsparameter θ_{RX} und θ_{GX} für Gen X als Realisierungen zweier unabhängiger identisch gamma-verteilter Zufallsgrößen angenommen. Deren Verteilung wird durch den Skalierungsparameter θ_0 und den Formparameter α_0 festgelegt.

Newtons Shrinkageparameter ist dann als θ_0 definiert.

3.2. Full-Bayesian Version des Gamma-Gamma Modells:

BUGS bietet die Möglichkeit das obige Modell sofort in einen einfachen Code oder einen Modellgraphen umzusetzen:

```
model
    {
        for (i in 1 : N) {
            R[i] ~ dgamma(alpha.1,theta.r[i])
            G[i] ~ dgamma(alpha.1,theta.g[i])
            theta.r[i]~dgamma(alpha.0,theta.0)
            theta.g[i]~dgamma(alpha.0, theta.0)
            rho[i] <- theta.g[i]/ theta.r[i]
                                    }
        alpha.1 ~ dgamma(1.0E-4,1.0E-4)
        alpha.0 ~ dgamma(1.0E-4,1.0E-4)
        theta.0 ~ dgamma(1.0E-4,1.0E-4)

    }
```

Dabei ist N die Anzahl der im Array untersuchten Gene. R und G sind die normierten Vektoren der gemessenen Rot / Grün Intensitäten. Die Vektoren theta.r und theta.g enthalten die wahren Expressionen für die einzelnen Gene, rho ist der Vektor mit den gesuchten Expressionsverhältnissen (ρ), die Parameter alpha.0 und alpha.1 entsprechen α_0 und α_1, theta.0 korrespondiert mit θ_0.

3.3. Vergleich des empirical Bayes *mit dem* full Bayes *Modell*

Newton bietet auf seiner Homepage vier Datensätze und eine Auswertungssoftware in Splus an. Diese Daten und Programme liegen in einem leicht modifizierten Splus Data-Dump vor. Mit ihnen lassen sich die Ergebnisse des Papers nachvollziehen (bis auf den Datensatz *Heat Shock*).

Die Orginaldaten liegen in den ASCII-Dateien mn1.csv, mn2.csv und mn1.csv vor (mn – Michel Newton). Die Splus Funktionen `mn.1.sfc` erstellt daraus den Datensatz `mn.1` (im Paper: *Control*). Analog werden `mn.2a` (im Paper: *IPTG-a*), `mn.2b` (im Paper: *IPTG-b*), `mn.3a` (im Paper: *Heat Shock*) und `mn.3b` von den entsprechenden Funktionen mn.**.sfc

erstellt. Das Wichtige an diesen Funktionen ist die Durchführung eines Normalisierungsalgorithmus und der Ausschluß von Genen mit einem Hintergrundrauschen, das größer als die gemessene Expression ist. Weiterhin werden die Daten in ein Listenformat gebracht, das die Daten in WinBUGS-Format überträgt. Die Komponenten der Liste sind xx (green), yy (red) und nn, die Bezeichnungen der analysierten Gene. Weiterhin muß im Code der Funktionen mn.**.sfc der Pfad zu den csv Dateien modifiziert werden.

Die Funktion `gg.fit.sfc` führt die Schätzungen im Gamma-Gamma Modell durch. Die folgende Tabelle gibt die Lösungen für die vorliegenden Daten wieder. Die Ergebnissse befinden sich auch in der Matrix `theta.gg`.

Datensatz	Bezeichnung	Anzahl NA Gene	α_1	α_0	ν
Kontrolle	mn.1	37	2.059682	2.098180	12.84157
Heat Shock	mn.3a	82	2.462559	1.511449	2.26905
IPTG-a	mn.2a	207	1.483641	1.845462	15.28732
IPTG-b	mn.2b	149	1.189896	1.571200	14.13351

Empirical Bayes Schätzer für das GG Modell

Diese Zahlen sind der Matrix `theta.gg` aus dem Splus-Data-Dump entnommen und stimmen bis auf die Ergebnisse zum Array mit den Heat Shock Genen mit den Angaben aus Tabelle 1 von Newton's Artikel überein. Der Parameter ν im IPTG-b Datensatz zeigt auch Abweichungen von Newton's Ergebnissen (Newton et al. 14.71, Tabelle: 14.13351) während die Parameter α_1 und α_0 bis auf Rundungsfehler keine Abweichungen zeigen.

Die Wirkung der Shrinkage wird durch die Splus-Funktion `shrink.plot.sfc` widergegeben. Die Abbildung zeigt Scatterplots, auf deren Achsen die $\log_{10}$ transformierten Intensitätsmessungen aufgetragen sind (x-Achse: Grün, y-Achse: Rot). Es werden für jedes Gen X die Punkte (G_X, R_X) und ($G_X+\nu$, $R_X+\nu$) durch eine Linie verbunden.

Im File gg_mn2a.odc ist ein *Doodle* für die *Full Bayesian* Version des Gamma-Gamma-Modells für den IPTG-a Datensatz gegeben. Zur Berechnung wurden 1000 Iterationen für den Run-In und 1000 Iterationen zum Samplen der Parameter ν (mu), α_1 (r.1) und α_0 (r.0) verwendet. Die Ergebnisse aus der MCMC-Prozedur sind:

node	mean	sd	MC	2.5%	median	97.5%	start	sample
mu	15.320	0.85830	0.1315	13.66	15.360	16.850	1001	1000
r.0	1.851	0.04783	0.0064	1.755	1.852	1.940	1001	1000
r.1	1.487	0.03603	0.0052	1.422	1.484	1.562	1001	1000

Diese Ergebnisse sind in guter Übereinstimmung mit denen aus dem *empirical Bayes* Ansatz.

4. Das Gamma-Gamma-Binomial Modell für Genexpression

In diesem Abschnitt soll der Frage nachgegangen werden, welche beobachteten Genexpressionsverhältnisse auf eine signifikante Expression schließen lassen. Aufgrund der Ergebnisse aus Abschnitt 3 kann dies noch nicht geleistet werden. Wir erhalten aus dieser Rechnung nur den Shrinkage-Faktor, der in die Schätzung des Expressionsverhältnisses

$$\hat{\rho}_{EB} = \frac{R + \upsilon}{G + \upsilon}$$

für einzelne Gene einfließt. Von Interesse ist im Folgenden die Schätzung des Anteils und eine Identifikation signifikant expremierter Gene. Hierzu schlagen Newton et al. die Verwendung eines Gamma-Gamma-Binomial Models vor.

4.1. Das Gamma-Gamma-Binomial Modell (GGB)

Das GGB Modell entsteht aus dem Gamma-Gamma Modell durch Hinzunahme einer dritten Modellschicht: Die unbekannte wahre Expressionsintensität von *Rot* und *Grün* ist zwischen einem gewissen Teil von Genen unterschiedlich (Anteil π mit $\mu_{RX} \neq \mu_{GX}$), für den Rest aber unverändert (mit $\mu_{RX} = \mu_{GX}$). Für Gene bei denen ein Unterschied vorliegt wird das Gamma-Gamma Modell verwendet: Die Skalierungsparameter θ_{RX} und θ_{GX} sind unabhängig identisch gamma-verteilt mit Shape-Parameter α_0 und Skalierungsparameter ν. Für Gene ohne einen Expressionsunterschied sind beide Messungen unabhängig identisch gamma-verteilt mit Skalierungsparameter θ_X, der unabhängig identisch gamma-verteilt ist mit Shape-Parameter α_0 und Skalierungsparameter ν.

Das Problem des GGB Modells liegt in der Unkenntnis über die Gene mit wirklicher unterschiedlicher Expression. Die Likelihood zur Schätzung der Parameter π, α_1, α_0, ν ist sehr komplex, da sie eine Summe über die 2^N (N – Anzahl der betrachteten Gene) mögliche Konfigurationen von (T_1, ..., T_N) ist, wobei T_X angibt ob bei Gen X eine unterschiedliche Expression (T_X=2) oder nicht (T_X=1) vorliegt. Newton et al. verwenden zur Lösung des Schätzproblemes einen EM-Algorithmus, der in der Funktion `ggb.fit.sfc` vorliegt. Die folgende Tabelle gibt die Schätzungen für die betrachteten 4 Datensätze wieder. Die Ergebnisse liegen auch in der Matrix `theta.ggb` vor.

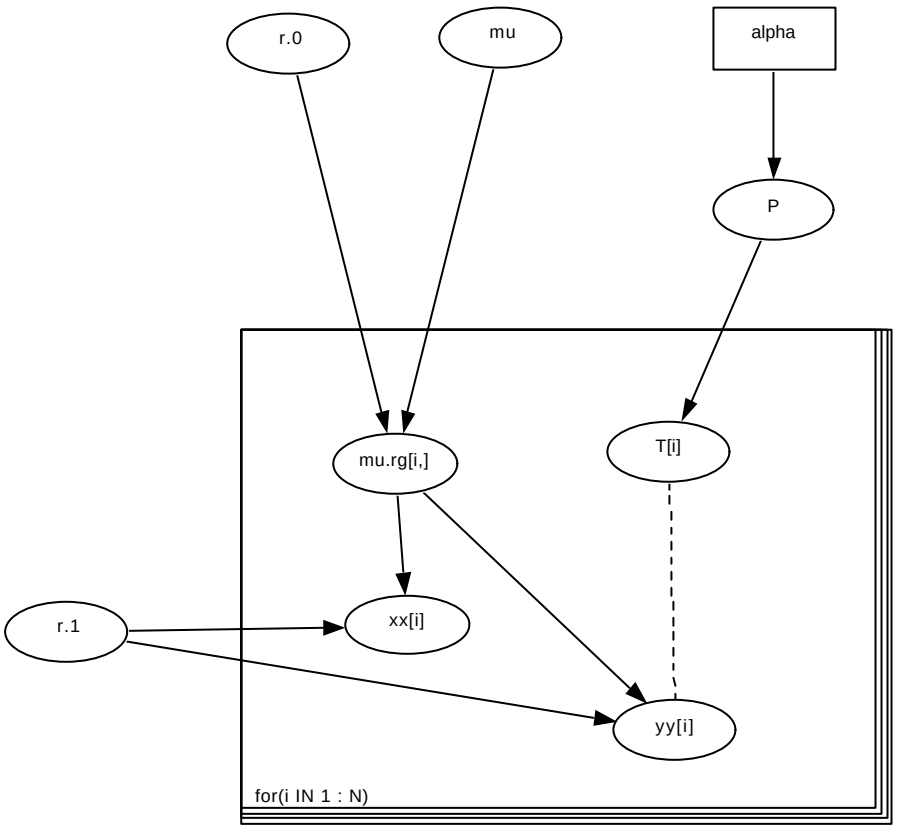
Datensatz	Bezeichnung	π	α_1	α_0	ν
Kontrolle	mn.1	22.904997	0.9438134	0.2847017	0.003048431
Heat Shock	mn.3a	3.236416	1.2558268	1.2773123	0.089082990
IPTG-a	mn.2a	12.534899	0.8162969	0.3706610	0.006879119
IPTG-b	mn.2b	9.688762	0.6785835	0.2843911	0.003589930

Empirical Bayes Schätzer für das GGB Modell

Bis auf den Heat Shock Datensatz liegen bis auf Rundungsfehler gute Übereinstimmungen mit Tabelle 2 aus dem Artikel vor.

4.2. Full-Bayesian Version des GGB Modells:

BUGS bietet auch für das GGB Modell die Möglichkeit es in einen einfachen Code oder einen Modellgraphen umzusetzen. Der Modellgraph ist in der folgenden Abbildung gegeben, darunter befindet sich der entsprechende Modell-Code.



Modell-Graph für das GGB-Modell.

Der Shape-Parameter für die Gammaverteilung von xx[i] ist mu.rg[i,1], der von yy[i] ist mu.rg[i,T[i]]. Somit wirkt im Falle keines Expressionsunterschiedes der gleiche Shape-

Parameter. Im Falle einer unterschiedlichen Expression wird der Skalierungs-Parameter von yy[i] als mu.rg[i,2] gesetzt. Der Form-Parameter ist für beide Farben und alle Gene identisch r.1 (α_1). Die Komponenten des Vektors mu.rg[i,] sind iid Realisierung einer gamma-verteilten Zufallsgröße mit Skalierungs-Parameter mu (θ) und Form-Parameter r.0 (α_0).

Es werden i=1, ..., N Gene betrachtet. Im Graphen sind xx[i] (Grün) und yy[i] (Rot) die für Gen i gemessenen Intensitäten. T[i] nimmt den Wert 1 an, wenn keine Expressions-unterschiede vorliegen und den Wert 2, falls Unterschiede bestehen. Die W'keit in die Kategorie T[i]=1 zu fallen, ist (1-p), die für Kategorie T[i]=2 ist p. Der Parameter alpha legt die Dirichletverteilung fest (in unserem einfachen Fall eigentlich eine Beta-Verteilung), welche als Apriori-Verteilung für p dient.

Nach einem Burn-In von 1000 Iterationen und weiteren 1000 Sampling-Schritten ergeben sich folgende Schätzungen für die Parameter im Datensatz IPTG-a (mn.2a):

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
P[1]	0.99370	0.0019	1.874E-4	0.9896	0.9938	0.9969	1001	1000
P[2]	0.00633	0.0019	1.874E-4	0.0031	0.0062	0.0104	1001	1000
r.1	12.4200	0.2542	0.041820	12.050	12.350	12.970	1001	1000
r.0	0.82090	0.0175	0.002068	0.7888	0.8210	0.8586	1001	1000
mu	0.37760	0.0159	0.002288	0.3474	0.3782	0.4091	1001	1000

Diese Zahlen sind in akzeptabler Übereinstimmung mit den *empirical Bayes*-Schätzern aus der Tabelle 2 der Arbeit von Newton et al.

Der Sampler-Output liefert eine Schätzung von 0.633% expremierter Gene, weiterhin das 95% Credibility-Intervall [0.0031; 0.0104] für den Anteil exprimierter Gene. Im IPTG-a Datensatz sind 207 der 4290 Gene nicht betrachtet. Somit werden von den 4083 verbleibenden relevanten Genen etwa 25 als exprimiert geschätzt mit einer Unsicherheit von 12 bis 42. Welche Gene sind es?

4.3. Identifikation der exprimierten Gene

Die Funktion `check.2.sfc` sucht unter den Genen der Datensätze IPTG-a (mn.2a), IPTG-b (mn.2b) und Heat Shock (mn.3a) die Gene heraus, die im Vergleich zu vorhergesagten Expressionen (aufgrund des GGB Modells) einen erhöhten LOD-Score besitzen (diese Rechnung wird von der Funktion `lod` durchgeführt). Im Datensatz IPTG-a (mn.2a) finden sich die 14 Gene:

```
> check.2.sfc()[[1]]

b4098   b4119   b0283   b0296   b0347   b4120   b1500
b2202   b0326   b2371   b0043   b0039   b2205   b1169
```

Zur Identifikation *exprimierter* Gene mit Hilfe der Gibbs-Sampler Prozedur, bietet sich das folgende Vorgehen an: Man samplet den Vektor T und erstellt dessen Statistik über die Standardausgabe **stat** in WinBUGS. Das ausgegebene Textfile läßt sich als ASCII File speichern und in Splus als Data-Frame einlesen. Das entsprechende Objekt ist `out.ggb.2a`. Von Interesse ist die Spalte `out.ggb.2a$mean` die die Mittelwerte der gesampelten Kategorien der Gene enthält. Aufgrund der Codierung von T (1-nicht exprimiert, 2-exprimiert) ist `out.ggb.2a$mean[i]-1` der Mittelwert der Aposteriori-Verteilung für die Expressionswahrscheinlichkeit von Gen i. Ein Umsortieren der Matrix `out.ggb.2a` nach der Größe der Aposteriori-Mittelwerte in abfallender Ordnung, würde die am Stärksten exprimierten Gene an die vorderen Positionen bringen. Im Folgenden sind die 30 in diesem Sinn am Stärksten exprimierte Gene gegeben:

```
dimnames(out.ggb.2a[sort.list((-1)*out.ggb.2a$mean),])[[1]][1:30]
```

```
b4098   b4119   b0283   b0296   b1500   b4120   b2202   b0043   b0347
b0039   b2371   b0326   b1169   b2205   b0602   b2204   b2673   b1389
b1673   b2203   b1998   b4247   b0770   b2055   b1018   b0669   b2197
b1912   b4354   b1506.
```

An welcher Stelle stehen, die von Newton gefundenen Gene in der obigen Reihe?

```
sort.winbugs.2a<-dimnames(out.ggb.2a[sort.list((-1)*out.ggb.2a$mean),])[[1]][1:30]
sort.newton.2a<-check.2.sfc()[[1]]
match(sort.newton.2a,sort.winbugs.2a)
1  2  3  4  9  6  5  7 12 11  8 10 14 13
```

Der Output der Splus- Funktion `match(sort.newton.2a,sort.winbugs.2a)` gibt an der i-ten Stelle die Position von `sort.newton.2a[i]` in `sort.winbugs.2a` an. Somit sind die 14 am stärksten exprimierten Gene in beiden Reihen gleich. Die vier am stärksten exprimierten Gene sind in beiden Reihen identisch. Nummer 9 bei Newton taucht als Nummer 5 im WinBUGS-Ergebnis auf, etc.

Newton et al. bestimmen die Anzahl exprimierter Gene durch die Betrachtung der aposteriori Odds für eine Expression des speziellen Genes. Die Funktion `odds.plot.sfc` gibt die Ergebnisse dieser Betrachtung graphisch wieder. Auf den Achsen sind $\log_{10}$-transformierten Intensitäten der beiden Farbkanäle aufgetragen (x-Achse: Grün, y-Achse: Rot). Somit stellt jeder Punkt eine beobachtete Intensitätskombination dar. In den Graphen sind drei Konturen für die Odds einer Expression eingetragen: 1:1, 10:1, 100:1. Von Interesse sind Gene mit einer Expressions-Odds von mindestens 1. Im Fall der IPTG-a Daten ist dies bei 14 Genen der Fall.

Im WinBUGS-Ergebnis würde ein Gen als *exprimiert* klassifiziert werden, falls seine aposteriori Expressionswahrscheinlichkeit mindestens 0.5 beträgt. Die Expressionswahrscheinlichkeiten der 20 am stärksten exprimierten Gene sind:

```
1.000 1.000 1.000 0.978 0.975 0.945 0.879 0.846 0.800 0.777 0.719 0.646
0.567 0.546 0.323 0.318 0.295 0.257 0.242 0.207
```

Somit ergeben sich auch im WinBUGS-Ansatz 14 exprimierte Gene. Dazu sind sie noch identisch mit den der *empirical Bayes* Prozedur.

5. Literatur

[Goldstein, 1995] Goldstein H, 1995, *Multilevel Statistical Models,* Wiley & Sons, New York, 1995

[Alizadeh et al., 2000] Alizadeh et al., 2000, *Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling,* Nature, 403, 503-511

[Golub et al., 1999] Golub TR et al., 1999, *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring,* Science, 286, 531-537

[Newton et al, 2000] Newton MA, Kendziorski CM, Richmond CS, Blattner FR, Tsui KW, 2000, *On differntial variablilty of expression ratios: Improving statistical inference about gene expression changes from microarray data,* Technical Report #139, Department of Biostatistics and medical Informatics, University of Wisconsin

Programme, Daten und Newton's Orginalartikel befinden sich im HM_HOM.zip, das von der Heidelberger Hompage http://www.biometrie.uni-hd.de/mb/techrep.htm heruntergeladen werden kann.