

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky



**Odhad relativní četnosti binomického
rozdělení pomocí klasického a
bayesovského přístupu v jazyce R**

BAKALÁŘSKÁ PRÁCE

Studijní program: [Data Analytics]

Autor: [Bc. Michal Lauer]

Vedoucí práce: [Ing. Ondřej Vilikus, Ph.D.]

Praha, Prosinec 2024

Poděkování

Děkuji svému vedoucímu za odborné vedení práce a průběžné konzultace a své přítelkyni za neocenitelnou podporu.

Abstrakt

Abstrakt.

Klíčová slova

Bayesovská statistika, odhad relativní četnosti, jazyk R

Abstract

Abstract.

Keywords

Bayesian statistics, relative frequency estimation, R language

Obsah

Úvod	9
1 Statistické metody	10
1.1 Inference	10
1.1.1 Problematika výběrových šetření	10
1.2 Frekventistická inference	10
1.2.1 Testování hypotéz	10
1.2.2 Metriky při testování hypotéz	10
1.2.3 Jednovýběrový odhad poměru s velkým vzorkem	10
1.2.4 Jednovýběrový odhad poměru s malým vzorkem	10
1.3 Bayesovská inference	11
2 Monte Carlo generování	12
2.1 Vyhodnocení generovaného rozdělení	12
2.1.1 Vyhodnocení hypotéz	12
2.1.2 Odhad poměru	12
3 Praktické odhady	13
3.1 Balíčky pro frekventistickou inferenci	13
3.1.1 Klasické test poměru	13
3.2 Software pro bayesovskou statistiku	13
3.2.1 Software BUGS	13
3.2.2 Balíček JAGS	15
3.2.3 stan	16
3.3 Simulace	17
3.3.1 Malý vzorek	17
3.3.2 Velký vzorek	17
3.3.3 Porovnání výsledků	17
Závěr	18
Použitá literatura	19
Použité balíčky	20
A Bayesovské modely	22

Seznam obrázků

Seznam tabulek

Seznam zdrojových kódů

Seznam použitých zkratek

BUGS Bayesian inference Using Gibbs

Sampling

Úvod

Tohle je **úvodní** *text*.

1. Statistické metody

Krátký úvod do historie, bayes, inferenční bayes (rozdělení) vs. inference (bod) citace Karla

1.1 Inference

proč to používáme, výběr vs. populace, reprezentativnost

1.1.1 Problematika výběrových šetření

reprezentativnost, definice populace, čas sběru, organizace sběru...

1.2 Frekventistická inference

Jak to funguje, jak to spoléhá na sampling distributions

1.2.1 Testování hypotéz

hladina významnosti, úroveň spolehlivosti, Testovací statistika, kritický obor, 1/2 stranný test p-hodnota, interval spolehlivosti

1.2.2 Metriky při testování hypotéz

Chyba I. a II. druhu, síla testu, velikost efektu

1.2.3 Jednovýběrový odhad poměru s velkým vzorkem

použití, předpoklady, poměrový Z test, binomický test, síla testu, velikost efektu

1.2.4 Jednovýběrový odhad poměru s malým vzorkem

Proč jsou důležité speciální metody, nějaké typy (wiki)

1.3 Bayesovská inference

Odvození bayesova vzorce, popis likelihood/aprior/data, druhy aprior/posterior

2. Monte Carlo generování

Halsing, Gibbs, HMC

2.1 Vyhodnocení generovaného rozdělení

korelace, ESS, monte carlo error...

2.1.1 Vyhodnocení hypotéz

Interval kredibility, ROPE, Bayesův faktor

2.1.2 Odhad poměru

3. Praktické odhady

3.1 Balíčky pro frekventistickou inferenci

3.1.1 Klasické test poměru

Test

test

```
stats::t.test()
```

test

Jednoduchý T-test

??

Simulace alfa = chyba 1. druhu

3.2 Software pro bayesovskou statistiku

3.2.1 Software BUGS

První software, který se snažil zpopularizovat bayesovské metody je software BUGS (*Bayesian inference Using Gibbs Sampling*). Projekt, jehož cílem bylo vytvořit software pro bayesovskou statistiku, započal už v roce 1989 na oddělení biostatistiky na univerzitě Cambridge a vedl k vytvoření programu BUGS. Postupem času se program vyvynul do nástroje WinBUGS, který je prezentován v této práci (The Bayesian Scientific Working Group, [2024](#)).

První verze programu byla představena na Bayesovské konferenci ve Valencii v roce 1991 a později byla distribuovaná na disketách. Z počátku byly implementovány s Gibbovým vzorkováním pouze jednoduché metody a úpravy, jako adaptivní vzorkování nebo bayesovské inverze. Po čase byla implementována i velmi limitující verze Metropolisova-Hastingova algoritmu, která fungovala na bázi mřížek ¹.

V 90. letech 20. století se projekt přesunul do Imperial College v Londýně a program BUGS se začal vyvíjet i pro osobní počítače se systémem Windows. Do verze WinBUGS se časem implementovala komplexní a plnohodnotná verze Metropolisova-Hastingova algoritmu, která dokázala pracovat bez aproximační tabulky. Velkou silou však bylo, že uživatel dokázal

¹Jedná se o tzn. *grid-search*, kdy se hledají správné hodnoty v předem definované tabulce.

definovat své apriorní předpoklady, data, vztahy a cykly v grafickém prostředí. Díky tomu mohli program WinBUGS používat i lidé bez zkušeností v programování.

V roce 2004 započala na univerzitě v Helsinách práce na projektu s názvem OpenBUGS, který měl za úkol splnit tři primární cíle:

- 1) rozdělit funkčnost softwaru od jejího vzhledu,
- 2) udělat verzi nezávislou na operačním prostředí, a
- 3) vytvořit experimentálnější prostředí pro zkoušení nových metod.

Pro splnění prvního cíle byl vyvynut nástroj s názvem BRugs. Díky oddělení funkčnosti a vzhledu lze BRugs napojit na další programy jako SAS, Excel nebo R a využívat tak simulací BUGS i mimo jeho prostředí. Ke splnění druhého cíle byla vyvynutá další verze softwaru bugs s názvem LinBUGS pro vývoj v linuxovém prostředí procesory Intel. Experimentální prostředí pro testování nových metod byla vytvořena open-source ² verze OpenBUGS. Vývoj obou aplikací se časem rozdělil a obě mají silné stránky v něčem jiném. Software OpenBUGS dokáže flexibilně měnit simulační metody nebo simulace provádět paralelně. Vývoj softwaru WinBUGS mířil primárně na excelenci v epidemiologickém a farmakokinetikou.

Aplikace z rodiny BUGS však trpěly několika nedostatky. Tím hlavním je, že jejich vývoj závisí na hrstce vývojářů a i přes to, že je kód (alespoň pro software OpenBUGS) veřejně dostupný, je vývoj problematický. To je způsobeno primárně komplexitou kódu a programovacím jazykem, ve kterém je software napsaný. Druhou velkou nevýhodou je právě programovací jazyk, který se nedokáže vyrovnat např. jazyku C++, ve kterém jsou napsána většina novodobých simulačních programů (Lunn et al., 2009).

V programovacím jazyce R lze s programy BUGS komunikovat pomocí hned několika balíčků. První možností jsou balíčky R2WinBUGS a R2OpenBUGS (Sturtz et al., 2005), které umí komunikovat s programy WinBUGS a OpenBUGS. Jelikož byl balíček OpenBUGS aktualizován naposledy v r. 2020, je smysluplnější³ v případě zájmu používat balíček R2WinBUGS, který poslední aktualizaci obdržel v r. 2024. Balíčky nenabízejí aplikační přístup, ale slouží jako automatizační pomůcka. Pokud funkcím předáte data, model a parametry simulací, tak se spustí zvolený software (OpenBUGS nebo WinBUGS), do kterého jsou automaticky zvoleny preference uživatel pomocí skriptu. Po dokončení se program sám zavře a výsledky jsou dostupné v jazyce R.

Druhou možností je balíček BRugs (Thomas et al., 2006), který dokáže komunikovat s OpenBUGS pomocí programatického prostředí a není tedy nutné, aby balíček/funkce na pozadí spouštěli samotné programy. Na rozdíl od balíčku R2WinBUGS, který externě spouští program, je BRugs více flexibilní. Nastavení probíhá pomocí metod, které nastavují, spouští a inicializují jednotlivé řetěze a simulace. Nevýhodou balíčku je, že je nutné ho kompilovat, což může být pro méně zkušené uživatele náročné.

²software, který má veřejný zdrojový kód; tzn. *otevřený software*.

³aktuální k 27. Sprnu, 2024.

Balíček {R2WinBUGS}

Výsledek

Odhad parametru p .

Posteriorní rozdělení jednotlivých chainů.

Vývoj jednotlivých chainů.

Autokorelace.

3.2.2 Balíček JAGS

2003 <https://www.jstor.org/stable/26447820>

JAGS - Just Another Gibbs Sampler

Balíček {rjags}

Tvorba modelu

Adaptační doba, která se volá automaticky.

Burn-in generování, je to pro každý chainu.

Generování vzorků z každého chainu.

Základní plot

Balíček {R2jags}

Možná úplně vynechat???

Divně spojený bugs and jags.

- Lze komplikovaně nastavit stejný seed

```
for (i in 1:n.chains) { init.values[[i]] <- inits[[i]] init.values[[i]].RNG.name <- RNGnameinit.values[[i]].RNG
<- runif(1, 0, 2^31) }
```

(asi by to šlo nastavit seed a pak to generovat setjně pomocí runif i nahoře)

- adapt = burnin nebo adapt = 100

```
if (n.burnin > 0) { n.adapt <- n.burnin } else { n.adapt <- 100 }
```

- Lze paralelizovat pomocí `jags.parallel`

Výsledky jsou pořád ze stejného posteriorního rozdělení a jsou validní, akorát se charakteristiky nerovnají.

Visualizace

Catterplot.

Posteriorní rozdělení.

Trace plot.

Autokorelace.

3.2.3 stan

Stan 1.0, 2012

http://www.stat.columbia.edu/~gelman/research/published/stan_jeps_2.pdf

Stan (není delší název).

aplikace, R implementace, výhody/nevýhody, používá hmc

Balíček {rstan}

Model.

```
set.seed(26)
x <- rbinom(10, 1, 0.6)

# stan.model <- rstan::stan(
#   model_code = readLines("./kapitoly/modely/stan.txt"),
#   model_name = "Jednoduchý příklad",
#   data = list(
#     N      = length(x), # Počet pozorování
#     x      = x,         # Vstupní data
#     alpha  = 0.1,       # Hodnota parametru alpha
#     beta   = 0.1        # Hodnota parametru beta
#   ),
#   init = list(
#     list(p = 0.5),
```



```
#      list(p = 0.5)
#      ),
#      chains = 2,
#      iter = 2000 + 5000,
#      warmup = 2000,
#      thin = 1,
#      seed = 59
#  )
```

3.3 Simulace

jak budou simulace provedné, jak budou vyhodnocené, nastavení ROPE/alternativ. pro odhad chyb

3.3.1 Malý vzorek

Bayes vs. vybraný vzorec vs. binomic

3.3.2 Velký vzorek

Bayes vs. vybraný vzorec vs. binomic

3.3.3 Porovnání výsledků

Jak testy dopadly

Závěr

Konec práce, závěr.

Použitá literatura

- Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009-11-10). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25), 3049–3067. <https://doi.org/10.1002/sim.3680>
- The Bayesian Scientific Working Group. (2024-08-26). *Bayesian Scientific Work Group – Bayesian Scientific Work Group*. <http://www.bayesianscientific.org/>

Použité balíčky

- Sturtz, S., Ligges, U., & Gelman, A. (2005). R2WinBUGS: A Package for Running WinBUGS from R. *Journal of Statistical Software*, 12(3), 1–16. <https://doi.org/10.18637/jss.v012.i03>
- Thomas, A., O’Hara, B., Ligges, U., & Sturtz, S. (2006). Making BUGS Open. *R News*, 6(1), 12–17. <https://cran.r-project.org/doc/Rnews/>

Přílohy

A. Bayesovské modely