

BAYESIÁNSKÁ ANALÝZA – CVIČENÍ 4

Toto cvičení je založeno na znalosti druhé části čtvrté kapitoly z učebnice Koop (2003): *Bayesian econometrics*, případně na odpovídající kapitole podkladového učebního textu *Bayesiánská analýza*.

Co bude náplní cvičení?

- ✎ Osvojení si metody importance sampling.

Zadání příkladů

1. *Importance sampling* Cílem tohoto úkolu je osvojení techniky importance sampling a pochopení vlastností tohoto postupu. Předpokládejme zcela jednoduchý model s jediným parametrem θ , jehož posteriorní hustota odpovídá $N(0, 1)$.
 - (a) Vytvořte program počítající posteriorní střední hodnotu a směrodatnou odchylku parametru θ pomocí Monte Carlo integrace.
 - (b) Vytvořte program počítající posteriorní střední hodnotu a směrodatnou odchylku parametru θ za použití techniky importance sampling a spočítejte rovněž střední hodnotu a směrodatnou odchylku použitých vah. Jako importance function použijte funkci hustoty pravděpodobnosti $t(0, 1, \nu)$.
 - i. Metoda importance sampling využívá generování vzorků ze známé kandidátské hustoty (importance function) a zahrnuje počítání vah jakožto podílů jader posteriorní hustoty a kandidátské hustoty vyhodnocené ve vygenerovaném kandidátovi. Výpočet středních hodnot funkcí parametrů, které nás zajímají pak vyžaduje provedení Monte Carlo integrace, kde místo aritmetických průměrů využíváme vážené průměry s vypočtenými vahami z „importance sampleru“.
 - (c) Proveďte Monte Carlo integraci a importance sampling pro různé hodnoty ν , např. $\nu = 2, 5$ a 100 pro daný počet replikací (např. $R = 10$). Porovnejte přesnost odhadů pro oba algoritmy a různou volbu ν . Všimněte se co se děje s vahami při zvyšujícím se ν .
 - (d) Zopakujte část (c) při využití $t(3, 1, \nu)$ jakožto importance function. Diskutujte faktory ovlivňující přesnost importance sampling ve světle skutečnosti, že importance function jen slabě aproximuje posteriorní hustotu.
 - (e) Vyzkoušejte i jiné importance function, např. $U(a, b)$, měňte i počet replikací R .
 - (f) Vypočítat požadované momenty posteriorní hustoty s využitím importance samplingu je vcelku jednoduchá záležitost. Existuje ale způsob, jak získat pomocí importance samplingu a s využitím umění generovat vzorky z importance funkce i přímo vzorky odpovídající posteriornímu rozdělení?
 - i. Myšlenka, jak na to není zas tak obtížná. Co máme k dispozici jsou jednak výběry z importance funkce, ale i velmi důležité váhy. Kdyby tedy existovala metoda, která by pro nějaký vzorek dat dokázala vybírat náhodné výběry, a to navíc i tak, že bychom mohli definovat s jakou pravděpodobností se ten který prvek původního výběru může vybrat, tak máme vyhráno.
 - ii. Touto metodou je metoda bootstrapu. V základním principu se nejedná o nic jiného, než o metodu, která vygeneruje z nějakého empirického rozdělení náhodný výběr (o požadované délce) tak, že pravděpodobnost výběru z každého vzorku je stejná. Máme-li tedy výběr o velikosti 100, každý z těchto prvků se bude vybírat s pravděpodobností 1 %, přičemž si sami můžeme zvolit, jestli se bude jednat o výběr s nahrazováním (to znamená, že po výběru nějakého prvku z empirického rozdělení tento prvek zůstává a v dalším kole může být vybrán znovu) prvku nebo bez nahrazování (když nějaký prvek vybereme, už jej v druhém kole vybrat nemůžeme). V našem případě využijeme samozřejmě variantu s nahrazováním, což je právě princip bootstrapu (druhý případ by odpovídal situaci, kdybychom potřebovali vybrat náhodnou podmnožinu výběru). Takovýto bootstrapový generátor si zvládneme vytvořit sami: stačí umět generovat čísla z uniformního rozdělení (pro 100 pozorování např. na intervalu 0

až 100, funkce `rand` transformovaná do požadovaného intervalu), která zaokrouhlíme na celá čísla (funkce `ceil`) a získáme tak index prvku, který z našeho vektoru `dat` máme vybrat. To opakuje tolikrát, kolik budeme chtít výběrů (počet výběrů nemusí odpovídat počtu prvků původního vektoru).

- iii. V našem případě ale nechceme vybírat prvky z kandidátské hustoty se stejnou pravděpodobností. Chtěli bychom předchozí proceduru modifikovat tak, aby pravděpodobnosti, s jakými budeme vybírat jednotlivé prvky vektoru byly proporcionální vahám získaným z `importance samplingu`. I zde bychom si příslušnou funkci mohli naprogramovat sami, např. tak, že si jednak nanormujeme váhy na jedničkový součet, vyhodíme prvky s nulovými vahami a vytvoříme nový umělý vzorek, ve kterém počet každého z prvků zvýšíme proporcionálně jeho váze (pravděpodobnosti). Následně pak na takto rozšířený vzorek aplikujeme metodu `bootstrapu` popsanou výše (i když každý prvek má stejnou pravděpodobnost výběru, jeho početní zastoupení zajistí požadovanou pravděpodobnost danou původními vahami).
- iv. Abychom si ale ulehčili práci, můžeme využít funkci `bootstrap` `Statistického toolboxu`. Tato funkce má různé podoby volání, ale my budeme potřebovat podobu:

```
[bootstat, bootsam] = bootstrap(1, [], theta_IS, weights, weight_IS);
```

kde `bootstat` by za normálních okolností obsahovalo bootstrapované statistiky či funkce původního vzorku, `bootsam` obsahuje bootstrapované indexy ze vzorku proměnných (v našem případě vektoru `theta_IS` (tyto indexy pak použijeme pro výběr z vektoru `theta_IS`), `[]` je prázdný argument, kde by za jiných okolností byla funkce nebo vektor funkcí aplikovaný na náš bootstrapovaný vzorek (např. bootstrapované střední hodnoty, směrodatné odchylky apod.), `theta_IS` je vektor parametrů z kandidátské hustoty a `weight_IS` je vektor odpovídajících vah z `importance samplingu`. `weights` je dodatečná volba funkce `bootstrap`, umožňující zavést požadované váhy. Jednička ve funkci pak říká, že chceme jeden jediný bootstrapový vzorek (o stejné velikosti jako vektor `theta_IS`). Kdybychom dali např. hodnotu 2, potom by se vytvořily dva vzorky a `bootsam` by byla matice indexů (o dvou sloupcích).