

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
TIKIMYBIŲ TEORIJOS IR SKAIČIŲ TEORIJOS KATEDRA

4 kurso 3 grupės studentas  
Kiprianas Spiridonovas

.....  
(parašas)

**BAJESINIS PASLĖPTŲ MARKOVO MODELIŲ PARAMETRŲ  
VERTINIMAS**

Bakalaurinis darbas

Vadovas  
prof. V. Kazakevičius

.....  
(parašas)

Vilnius  
2014



## Turinys

|  |    |
|--|----|
| Išvadas .....  | 5  |
| 1. Modelis ir metodai .....  | 7  |
| 1.1. Paslėptieji Markovo modeliai .....                            | 7  |
| 1.2. Bajesinis parametrų vertinimas .....                          | 7  |
| 1.3. Monte-Karlo Markovo grandinės.....                            | 8  |
| 1.3.1. Metropolis-Hastings algoritmas.....                         | 8  |
| 1.3.2. Gibbs imčių algoritmas.....                                 | 9  |
| 1.3.3. Imties kokybė ir konvergavimo tikrinimas .....              | 10 |
| 1.4. Normaliųjų skirstinių mišinio paslėptas Markovo modelis ..... | 11 |
| 1.4.1. Gibbs algoritmo pritaikymas .....                           | 12 |
| 1.4.2. Algoritmo įgyvendinimas kompiuteryje .....                  | 13 |
| 1.5. Palyginimas su Baum-Welch algoritmu .....                     | 14 |
| 2. Rezultatai .....  | 15 |
| Išvados .....  | 21 |
| Summary .....  | 23 |
| Literatūros sąrašas .....  | 25 |
| Priedas Nr. 1 .....  | 27 |



## Įvadas

Šio darbo tikslas yra įgyvendinti ir ištirti būdą įvertinti paslėpto Markovo modelio parametrų skirstinius naudojant Bajesinės statistikos metodus. Gautus parametrų skirstinius galima naudoti išvadoms daryti arba procesams palyginti.

Paprastoje Markovo grandinėje kiekvieną būseną galima stebėti tiesiogiai, taigi kurį laiką stebint procesą jo perėjimo matricą (diskrečiuoju atveju) galima sudaryti suskaičiavus būsenų pasikeitimus. Paslėptame Markovo modelyje (angl. *hidden Markov model*) būsenos tiesiogiai stebėti negalime, tačiau galime stebėti kitą atsitiktinę seką, kurios skirstinys kiekviename žingsnyje priklauso nuo grandinės būsenos tuo metu. Toks modelis yra kur kas bendresnis ir pritaikomas daugeliui praktinių uždavinių, pavyzdžiui, balso ir gestų atpažinimui.

Jeigu turime normalių skirstinių mišinį, kur kiekvienos reikšmės priklausomybę populiacijai aprašo nestebima Markovo grandinė, tai yra paslėptas Markovo modelis. Tokį atvejį ir nagrinėsime.

Turint stebėtą seką galima iškelti įvairius uždavinius, pavyzdžiui, kokia yra tikėtiniausia būsena tam tikrame žingsnyje nepriklausomai nuo kitų, kokia yra tikėtiniausia paslėptų būsenų *seka*, kokie yra modelio parametrai. Spręsime pastarąjį: ieškosime paslėptos grandinės tikimybių, o taip pat mišinio populiacijų vidurkių ir dispersijų.

Dažniausiai tam yra naudojamas Baum-Welch algoritmas, kuris veikia *expectation maximization* principu iteratyviai pagerindamas parametrų įverčius, kol pasiekia tikėtinumo funkcijos kritinį tašką – įprastai lokalų maksimumą. Nagrinėsime alternatyvą jam, kuri yra paremta Gibbs algoritmu ir paskaičiuoja ne tik taškinis įverčius, bet ir imtis iš parametrų aposteriorinių skirstinių. Imtis galima naudoti gautų įverčių kokybei įvertinti, o taip pat statistinių išvadų gavimui.



# 1. Modelis ir metodai

## 1.1. Paslėptieji Markovo modeliai

Paslėptasis Markovo modelis susideda iš poros atsitiktinių kintamųjų sekų  $(X_n, Y_n)$ , kur kiekvienas  $X_n$  priima diskrečias vertes ir jo pasiskirstymas priklauso tik nuo ankstesnio sekos nario  $X_{n-1}$ , t.y.

$$p(X_n | X_1, X_2, \dots, X_{n-1}) = p(X_n | X_{n-1}).$$

$X_n$  yra diskreti homogeninė Markovo grandinė, taigi ją pilnai aprašo pradinis skirstinys  $p_0$  ir perėjimo matrica  $T$ . Tariaime, kad  $X_n$  įgyjamų reikšmių stebėti negalime.

Kiekvieno  $Y_n$  pasiskirstymas priklauso tik nuo  $X_n$  ir jų reikšmes galime stebėti.

Nagrinėsime tokį modelį, kur  $Y_n$  – normaliai pasiskirstę dydžiai, kurių vidurkis ir dispersija priklauso nuo paslėptos būsenos. Tarkime, duoti  $(\mu_1, \dots, \mu_k)$  ir  $(\sigma_1, \dots, \sigma_k)$ . Tada

$$Y_n \sim N(\mu_{X_n}, \sigma_{X_n}^2).$$

Tokiu atveju stacionarusis  $Y_n$  skirstinys yra atitinkamų normalių skirstinių mišinys, pasvertas pagal  $X_n$  būsenų stacionarias tikimybes.

Reikia turint tik stebėtą  $Y_n$  seką nustatyti jos skirstinio parametrus, o taip pat  $X_n$  grandinės parametrus (pradines tikimybes ir perėjimo matricą).

## 1.2. Bajesinis parametrų vertinimas

Nagrinėkime modelį  $Y \sim F(\theta)$ , kur  $F$  – žinomas skirstinys, o  $\theta$  – nežinomas parametrų vektorius. Tarkime, kad  $y$  yra stebėtas  $Y$  reikšmių vektorius.

$\theta$  galime priskirti apriorinį pasiskirstymą, nepriklausantį nuo  $y$ , kuris atspindi mūsų išankstinę nuomonę\* apie jo galimas reikšmes.

Tada pagal Bajeso teoremą galime gauti anaujintą aposteriorinį skirstinį, kuris atspindi  $\theta$  reikšmių tikimybes atsižvelgiant į stebėtus  $y$ :

$$p(\theta | y) = \frac{p(y | \theta)p(\theta)}{p(y)}.$$

Paprastais atvejais šiuos skirstinius galima suskaičiuoti tiesiogiai. Pavyzdžiui jeigu  $Y \sim \text{Bernoulli}(q)$  su aprioriniu skirstiniu  $q \sim \text{Beta}(1, 1)$  ir iš atliktų  $n$  bandymų  $k = \sum_{i=1}^n y_i$  buvo „sėkmingi“, tai aposteriorinis skirstinys bus  $q \sim \text{Beta}(1 + k, 1 +$

---

\*Jeigu išankstinės nuomonės neturime, yra būdų pasirinkti neinformatyvius skirstinius, pavyzdžiui, Jeffreys prior.

$n - k$ ), nes

$$p(q|k, n) = \frac{p(k|q, n)p(q)}{p(k|n)} = \frac{q^k(1-q)^{n-k} \cdot 1}{\int_0^1 q^k(1-q)^{n-k} dq}.$$

Jeigu dar atliktume  $n_1$  bandymų su  $k_1$  „sėkmingų“, tai naujasis pasiskirstymas būtų  $q \sim \text{Beta}(1+k+k_1, 1+n+n_1-k-k_1)$ , t.y. skirstinio išraiška liktų ta pati, tik su atnaujintais parametrais. Dėl to Beta skirstinys yra vadinamas jungtiniu Bernulio ir binominiu skirstiniais.

Sudėtingesniems modeliams įprastai negalioja šita savybė ir rasti tikslas aposteriorinių skirstinių išraiškas pasidaro nepraktiška. Tačiau jeigu galėtume iš skirstinio gauti imtį, tai galėtume ją naudoti apytiksliai nustatyti skirstinio formai ir dominančioms statistikoms. Vienas iš būdų netiesiogiai gauti imtį iš skirstinio yra Monte-Karlo Markovo grandinės.

### 1.3. Monte-Karlo Markovo grandinės

Monte-Karlo Markovo grandinė yra Markovo grandinė  $X_n$ , specialiai sukonstruota taip, kad jos stacionarusis skirstinys  $\pi$  sutaptų su ieškomu skirstiniu  $p$ . Tai leidžia simuliuojant grandinę  $X_n$  gauti imtį iš skirstinio  $p$ , kuris gali būti per sudėtingas tiesioginiam skaičiavimui. Tačiau Markovo grandinėje gretimi  $X_n$  yra priklausomi, taigi jų gali prireikti daugiau, nei tiesiogiai imant vertes iš  $p$ .

Yra įvairių būdų sudaryti tokią grandinę.

#### 1.3.1. Metropolis-Hastings algoritmas

Tai labai bendras Monte-Karlo Markovo grandinės sudarymo būdas. Jį 1953-iaisiais metais aprašė Nicholas Metropolis ir kiti [MRR<sup>+</sup>53], o 1970-aisiais apibendrino W. K. Hastings [Has70].

Tarkime galime žinome funkciją  $f(x)$ , kuri yra proporcinga ieškomo skirstinio tankio funkcijai  $p(x)$ . Taip pat pasirinkime kokį nors skirstinį  $q(x|y)$ , vadinamą kandidatų skirstiniu. Išsirinkime bet kokį pradinį  $x_0$ .

Tarkime duotas  $x_i$ . Pagal skirstinį  $q(x'|x_i)$  atsitiktinai pasirenkame kandidatą  $x'$ . Tegu

$$\alpha = \min \left\{ 1, \frac{f(x')q(x_i|x')}{f(x_i)q(x'|x_i)} \right\}.$$

Santykis  $f(x')/f(x_i) = p(x')/p(x_i)$  atspindi kiek kandidatas  $x'$  yra labiau tikėtinas, nei dabartinė vertė  $x_i$ . Santykis  $q(x_i|x')/q(x'|x_i)$  reikalingas, jeigu  $q$  nėra simetrinis.



Tada sekančią vertę pasirenkame pagal

$$x_{i+1} = \begin{cases} x' & \text{su tikimybe } \alpha \\ x_i & \text{su tikimybe } 1 - \alpha. \end{cases}$$

Algoritmo konvergavimo greitis priklauso nuo  $q$  pasirinkimo. Geriausi rezultatai gaunami, kai jo forma panaši į  $p$ , nes tada yra dažniau priimami kandidatai, tačiau praktikoje yra naudojami paprasti skirstiniai.

Pats paprasčiausias atvejis yra  $q(x)$  kuris nepriklauso nuo ankstesnio žingsnio. Jeigu  $p(x)$  suportas yra baigtinis, tai galima tiesiog naudoti uniforminį skirstinį.

Dažnai naudojamas  $q(x|y) = N(x|y, \sigma^2)$ . Tada  $x_i$  trajektorija priklauso nuo  $\sigma$  pasirinkimo: jeigu jis per mažas, tai aplankyti visą erdvę gali prireikti labai daug žingsnių, o jeigu per didelis tai dideli šuoliai gali neatitikti  $p$  formos ir dėl to su mažais  $\alpha$  retai priimti kandidatai.

### 1.3.2. Gibbs imčių algoritmas

Jeigu reikalinga imtis iš daugiamačio skirstinio  $p(x) = p(x_1, \dots, x_n)$ , iš kurio imti tiesiogiai negalime, tačiau galime imti iš  $p(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$  su kiekvienu  $i = 1, \dots, n$ , tai galime naudoti Gibbs algoritmą.

Kaip ir Metropolis-Hastings atveju, pradedame nuo bet kokio  $x^0$  (žingsnio numerį žymėsime viršutiniais indeksais).

Tada duotam  $x^i$  kiekvieną komponentę atnaujiname iš eilės:

$$\begin{aligned} x_1^{i+1} &\sim p(x_1|x_1^i, \dots, x_n^i) \\ x_2^{i+1} &\sim p(x_2|x_1^{i+1}, x_3^i, \dots, x_n^i) \\ &\dots \\ x_j^{i+1} &\sim p(x_j|x_1^{i+1}, \dots, x_{j-1}^{i+1}, x_{j+1}^i, \dots, x_n^i) \\ &\dots \\ x_n^{i+1} &\sim p(x_n|x_1^{i+1}, \dots, x_{n-1}^{i+1}) \end{aligned}$$

Kai turime visas naujo  $x^i$  komponentes, tuos žingsnius pakartojame, kad gautume  $x^{i+1}$ , ir t.t.

Šį algoritmą 1984-aisiais metais aprašė S. Geman ir D. Geman [GG84]. Kai reikalavimai patenkinti, t.y. kiekvienam parametrui turime sąlyginius skirstinius priklausomai nuo kitų, tai įprastai šis metodas įprastai konverguoja greičiau, nei Metropolis-Hastings. Jeigu sąlyginius skirstinius turime tik daliai parametrų, tai galima naudoti abiejų algoritmų kombinaciją.

### 1.3.3. Imties kokybė ir konvergavimo tikrinimas

Jeigu  $\{x_n\}$  yra Monte-Karlo Markovo grandinė, tai jos stacionarus skirstinys  $\pi$  turėtų artėti į  $p$ , kai  $n$  artėja į begalybę. Gibbs algoritmo atveju sąlygos tam yra labai bendros: jei grandinės granduolys  $T$  yra  $\pi$ -neredukuojamas,  $\pi T = \pi$  ir  $T$  – neperiodinis, tai  $\pi$ -beveik visiems  $x$  galioja  $\|T^n(x, \cdot) - \pi\| \rightarrow 0$  [Tie94]. Šitos sąlygos galioja daugumai praktiškai pasitaikančių uždavinių, tačiau tai nieko nepasako apie konvergavimo greitį.

Pirmasis sekos narys pasirenkamas neatsižvelgiant į  $\pi$ , taigi iš pradžių grandinės reikšmės neatitiks  $\pi$  skirstinio ir tik po kurio laiko prie jo priartės. Dėl to įprastai tam tikra dalis pirmųjų reikšmių yra nenaudojama. Angliškai ta dalis yra vadinama *burn-in period*. Kiek iteracijų atmesti dažnai galima nustatyti vizualiai iš grafiko, arba tiesiog iš anksto pasirenkama atmesti pavyzdžiui 20% ir jei grandinė pakankamai ilga to įprastai užtenka, tačiau yra ir formalių būdų tam kiekiui nustatyti. Pavyzdžiui, normaliai pasiskirsčiusioms vertėms, statistika  $R$  iš [FS14] artėja prie 1, kai skirstinys konverguoja į stacionarųjį. Praktiškai jai suskaičiuoti yra susimuliuojamos kelios grandinės (su skirtingomis atsitiktinių skaičių sekomis) ir kiekviena jų padalijama į vienodo ilgio blokus. Tada kiekviename bloke lyginami grandinių skirstiniai, vadovaujantis prielaida, kad grandinių skirstinių supanašėjimas reiškia konvergavimą į stacionarųjį skirstinį. [FS14] tam naudojamas kiekvienos grandinės dispersijos santykis su bendra dispersija per visas grandines, tačiau tai gali būti ne labai tinkama skirstiniams, kurie nėra normalūs. Panašiai palyginimui galima būtų taikyti Kolmogorovo-Smirnovo arba, diskrečiuoju atveju,  $\chi^2$  statistiką.

Monte-Karlo Markovo grandinės naudojimo tikslas yra gauti tam tikro dydžio  $n$  nepriklausomų reikšmių imtį iš jos stacionariojo skirstinio, tačiau grandinės nariai yra priklausomi vienas nuo kito. Tolygiuoju atveju tai galima įvertinti per autokoreliaciją

$$\rho_k = \frac{\text{Cov}(x_i, x_{i+k})}{\text{Var}(x_i)}.$$

Norint sumažinti gretimų narių koreliaciją galima naudoti ne kiekvieną sekos narį, o tik kas  $m$ -ąjį. Tai vadinama grandinės praretinimu (angl. *thinning*). Tarkim, jei  $\rho_k \approx \rho_1^k$  ir norime, kad būtų  $\rho_1 < \rho'$ , tai  $m = \lceil \frac{\ln \rho'}{\ln \rho_1} \rceil$ . Praktikoje autokoreliacija nebūtinai neigiamai įtakoja imties reprezentatyvumą, tačiau praretinimas gali būti naudingas, nes tada sekos apdorojimas ir vizualizavimas reikalauja mažiau kompiuterio resursų ir tai leidžia simuliuoti ilgesnę grandinę.

Bendru atveju imties kokybei įvertinti galima naudoti efektyvaus imties dydžio įvertį

$$ESS = \frac{n}{1 + 2 \sum_{k=1}^{\infty} \rho_k},$$

kuris atsižvelgia į visus  $\rho_k$ . Jis pagrįstas tuo, kad jei turime seką nepriklausomų

$X_1, X_2, \dots$  pasiskirsčiusių pagal  $p$  ir  $\text{Var}(X_k) = \sigma^2$ , tai

$$\text{Var} \left( \frac{1}{n} \sum_{k=1}^n X_k \right) = \frac{\sigma^2}{n},$$

o tada su  $x_k$  galioja

$$\text{Var} \left( \frac{1}{n} \sum_{k=1}^n x_k \right) \approx \frac{\sigma^2}{ESS}.$$

#### 1.4. Normaliųjų skirstinių mišinio paslėptas Markovo modelis

[RG97] yra aprašytas Bajesinis modelis normaliųjų skirstinių mišiniam. [Ryd08] pritaiko panašų modelį tam atvejui, kai priklausomybę populiacijai nustato nestebima Markovo grandinė ir visos populiacijos turi bendrą dispersiją  $\sigma^2$ . Apibendrinsime šį modelį atskiroms ir galimai skirtingoms dispersijoms pagal [RG97].

Tegu  $(X, Y)$ ,  $X = (X_1, \dots, X_N)$ ,  $X_k \in \{1, \dots, M\}$ ,  $Y = (Y_1, \dots, Y_N)$  – paslėptas Markovo modelis, toks kad

$$Y_k \sim N(\mu_{X_k}, \sigma_{X_k}^2).$$

Tegu  $T - X$  grandinės perėjimo matrica, o  $p = (p_1, \dots, p_M)$  – jos stacionarusis skirstinys. Tariaime, kad  $X_1$  būseną pasirenkama pagal  $p$ .

Turint stebėtą  $Y$  seką  $y = (y_1, \dots, y_N)$  reikia nustatyti parametrus  $p$ ,  $T$ ,  $\mu = (\mu_1, \dots, \mu_M)$  ir  $\sigma = (\sigma_1, \dots, \sigma_M)$ . Galime jiems apibrėžti apriorinius skirstinius.

$T$  yra eilutėse stochastinė matrica, taigi kiekvienai jos eilutei galime priskirti Dirichlė skirstinį, ir tą patį padaryti su  $p$ :

$$(T_{i,1}, \dots, T_{i,M}) \sim \text{Dir}(1, 1, \dots, 1),$$

$$p \sim \text{Dir}(1, 1, \dots, 1).$$

Mišinių atveju aprioriniai skirstiniai negali būti visiškai neinformatyvūs, nes tokiu atveju aposterioriniai skirstiniai būtų nekorektiški [RG97, 2.4 skyrius].

Vidurkiams  $\mu$  galima būtų pasirinkti nepriklausomus normalius skirstinius su pakankamai didelėmis dispersijomis, kad minimaliai įtakoti rezultatus. Jeigu turime pakankamai daug stebėtų reikšmių, tai tikėtina, kad populiacijų vidurkiai nėra labai nutolę nuo stebėtų reikšmių intervalo. Tegu  $\mu_i \sim N(\xi, (aR)^2)$ , kur  $\xi = \frac{\min y + \max y}{2}$ ,  $R = \max y - \min y$ . Tokiu atveju skirstinys centruotas ties stebėtų reikšmių intervalo centru ir turi  $a$  kartų didesnę standartinę nuokrypį negu intervalo plotis. Pasirenkame  $a = 10$ .

Standartiniais nuokrypiams pasirenkame

$$\begin{aligned}\sigma_i^{-2} &\sim \Gamma(\alpha, \beta), \\ \beta &\sim \Gamma(g, h).\end{aligned}$$

Tada nustatę  $g < 1 < \alpha$  galime formalizuoti nuomonę, kad  $\sigma_i$  yra tarpusavyje panašūs. Nustatome  $\alpha = 2$ ,  $g = 0.2$ ,  $h = (aR)^{-2}$ .

#### 1.4.1. Gibbs algoritmo pritaikymas

Kiekvienam parametrui galime aprašyti sąlyginį skirstinį nuo likusių parametrų ir paslėptos grandinės būsenų  $X_k$ :

$$\begin{aligned}p|\dots &\sim \text{Dir}(1 + C_1, 1 + C_2, \dots, 1 + C_M), \\ (T_{i,1}, \dots, T_{i,M})|\dots &\sim \text{Dir}(1 + C_{i,1}, 1 + C_{i,2}, \dots, 1 + C_{i,M}), \\ \mu_i|\dots &\sim \text{N}\left(\frac{S_i + \kappa\xi\sigma_i^2}{n_i + \kappa\sigma_i^2}, \frac{\sigma^2}{n_i + \kappa\sigma_i^2}\right), \\ \sigma_i^{-2}|\dots &\sim \Gamma\left(\alpha + \frac{1}{2}C_i, \beta + \frac{1}{2}\sum_{k|X_k=i} (y_k - \mu_i)^2\right), \\ \beta &\sim \Gamma(g + M\alpha, h + \sigma^{-2}).\end{aligned}$$

Čia  $C_i = \#\{X_k = i | k = 1, \dots, N\}$ ,  $C_{i,j} = \#\{X_k = i, X_{k+1} = j | k = 1, \dots, N-1\}$ .

Jeigu fiksuojame parametrų  $p$ ,  $T$ ,  $\mu$ ,  $\sigma$  vertes, tai pagal Bajeso teoremą galime paskaičiuoti pirmosios grandinės būsenos tikimybes atsižvelgiant į stebėtus duomenis:

$$\begin{aligned}P(X_1 = j|y) &\propto P(y|X_1 = j)P(X_1 = j) \\ &= p_j P(y|X_1 = j) \\ &= p_j \text{N}(y_1|\mu_j, \sigma_j^2) P(y_{2:N}|X_1 = j),\end{aligned}\tag{1}$$

Čia  $y_{a:b} = (y_a, y_{a+1}, \dots, y_b)$ , o  $a \propto b$  reiškia, kad  $a$  yra proporcingas  $b$ . Kadangi diskrečių būsenų yra baigtinis skaičius, tai tikimybes iš proporcingų skaičių galime gauti normalizavę taip, kad jų suma būtų lygi 1.

Panašią išraišką gauname ir perėjimo iš vienos būsenos į kitą tikimybei:

$$\begin{aligned}P(X_k = j|y_{k:N}, X_{k-1} = i) &\propto P(y_{k:N}|X_k = j, X_{k-1} = i)P(X_k = j|X_{k-1} = i) \\ &= T_{i,j} \text{N}(y_k|\mu_j, \sigma_j^2) P(y_{k+1:N}|X_k = j),\end{aligned}\tag{2}$$

Tikėtinumą  $P(y_{2:N}|X_1 = j)$  galime suskaičiuoti rekursyviai pagal

$$P(y_{k:N}|X_{k-1} = i) = \sum_j T_{i,j} N(y_k|\mu_j, \sigma_j^2) P(y_{k+1:N}|X_k = j),$$

$$P(y_N|X_{N-1} = i) = \sum_j T_{i,j} N(y_N|\mu_j, \sigma_j^2).$$

Taigi paslėptų būsenų tikimybės atsižvelgiant į stebėtus duomenis yra nehomogeninė Markovo grandinė.

Igyvendinant Gibbs algoritmą iš pradžių sugeneruojame parametrus pagal jų nepriklausomus apriorinius skirstinius, o tada pakaitomis generuojame paslėptų būsenų grandinę sąlyginai nuo parametrų, ir parametrus sąlyginai nuo gautos grandinės. Jeigu stebėtas daugiau nei vienas  $y$  vektorius (su galimai skirtingais ilgiais), tai paslėptą grandinę galima simuliuoti su kiekvienu jų, vienu po kito, ir sumuoti gaunamas statistikas  $(C_i, C_{i,j}, \sum_{k|X_k=i} (y_k - \mu_i)^2)$ .

Rezultate gaunama imtis iš *a posteriori* parametrų skirstinio  $P(\theta|y)$ . Vizualiai nubraižius skirstinį galima nustatyti ar gauti rezultatai patikimi. Jeigu parametro reikšmės siaurai susikoncentravusios apie vieną tašką, tai įvertis turbūt patikimas, tačiau jeigu yra platus multimodalinis pasiskirstymas, tai gali būti kad išvadoms daryti trūksta duomenų.

Taškinius įverčius parametrų galima gauti įvairiomis statistikomis. Mes naudosime modą. Jei  $x$  – parametrų vektorius, kurio imtį turime, tai apytikslę modą pažymėkime  $\hat{x}$ . Ją galima gauti naudojant *shorth* metodą iš [ABH72].

Taip pat iš skirstinių galima gauti didžiausio tankio intervalus (*highest density interval*, HDI) ir juos naudoti daryti statistinėms išvadoms apie parametrų reikšmes. Pavyzdžiui, jeigu turime dvi aibes stebėtų duomenų iš galimai skirtingų procesų, tai joms abiejoms gavę parametrų skirstinius galime rezultatus palyginti ir paskaičiuoti tikimybę, kad jas sukūrė tas pats procesas. Naudosime didžiausio tankio intervalo gavimo būdą, kuris yra aprašytas [CS99]. Pažymėkime jo apatinę ir viršutinę ribas atitinkamai  $L_x$  ir  $U_x$ . Tai galima būtų tiesiogiai taikyti  $\mu$  ir  $\sigma$ , tačiau ne  $p$  ar  $T$  eilutėms, nes jų nariai priklauso vienas nuo kito (suma turi būti lygi 1) ir į tai reikia atsižvelgti.

#### 1.4.2. Algoritmo įgyvendinimas kompiuteryje

Gibbs imčių algoritmas aprašyto paslėpto Markovo modelio parametrų nustatyti įgyvendintas C++11 kalba. Taip pat panaudota Armadillo tiesinės algebros biblioteka.

Atsitiktinių skaičių generavimui panaudotos standartinės bibliotekos funkcijos su Mersenne-Twister 64-bit pseudoatsitiktinių skaičių generatoriumi. Pradinę jo būseną

(*seed*) galima nurodyti kaip parametą, taigi gauti rezultatai yra deterministiniai ir atkartojami.

Skaičiuojant (1) ir (2) vertes gali būti gaunami skaičiai labai artimi nuliui ( $< 10^{-1000}$ ), nes rekursyviai sudauginama ir sumuojama daug tikėtinumų, kurie gali būti labai maži, o normalizacija atliekama tik pabaigoje. Įprastas 64-bitų slankiojo kabelio formatas (*double* tipas) netinkamas saugoti tokiems mažiems skaičiams. Galime saugoti ir manipuluoti jų logaritmus, tačiau tada reikia ypatingo būdo skaičiuoti sumas. Tarkime turim  $b_i = \ln a_i$ . Tada

$$\ln \sum_i a_i = \ln \sum_i e^{b_i} = \ln e^C \sum_i e^{-C} e^{b_i} = C + \ln \sum_i e^{b_i - C}.$$

Jeigu pasirenkame  $C = \max_i b_i$ , tai  $e^{b_i - C}$  nebus per maži *double* tipui ir juos galima susumuoti kompiuteryje.

## 1.5. Palyginimas su Baum-Welch algoritmu

Įprastai paslėptų Markovo modelių parametrų įverčiams gauti naudojamas Baum-Welch algoritmas. Jis gražina taškinis lokaliai didžiausio tikėtinumo įverčius.

Jis veikia iteratyviai. Pradedama nuo atsitiktinai spėtų (arba *a priori* numanomų) pradinių parametrų reikšmių  $\theta = (p, T, \dots)$ . Tada naudojant tiesioginės-atbulinės eigos algoritmą paskaičiuojamos tikimybės

$$\begin{aligned}\alpha_i(k) &= P(y_{1:k}, X_k = i | \theta), \\ \beta_i(k) &= P(y_{k:N} | X_k = i, \theta).\end{aligned}$$

Skaičiavimas vyksta rekursyviai, panašiai kaip ir atitinkamas žingsnis Gibbs algoritme. Tada pagal tam tikras taisykles galima gauti atnaujintus įverčius parametrų  $\theta'$ . [BPSW70] parodyta, kad  $P(\theta' | y) \geq P(\theta | y)$ , o lygybė galioja tik tada, kai pasiektas tikėtinumo funkcijos kritinis taškas.

Kai kuriais atvejais tikėtinumo funkcija gali turėti daug kritinių taškų, taigi algoritmas kartojamas su įvairiomis pradinėmis reikšmėmis, o paskui išrenkamas geriausias rezultatas.

Jeigu norima gauti ne tik taškinį įvertį, bet ir pasikliautinąjį intervalą, galima naudoti pakartotinės atrankos (*resampling*) metodus sudaryti naujoms imtims ir kiekvienai jų taikyti šį algoritmą.

## 2. Rezultatai

Tiriamą metodo veikimui įvertinti buvo atsitiktinai sugeneruotos duomenų aibės su įvairiais parametrais. Panaudota pora perėjimo tikimybių matricių.

$$S_M = (s_{i,j})$$

$$s_{i,j} = \begin{cases} \alpha & \text{kai } i = j, \\ \frac{1-\alpha}{M-1} & \text{kai } i \neq j. \end{cases}$$

kur  $i, j = 1, 2, \dots, M$ , o  $M \in \{2, 3, 4\}$  yra būsenų skaičius, o  $\alpha = 0.9$ . Pagal šią matricę su tikimybe  $\alpha$  pasilieka esamoje būsenoje, o priešingu atveju pereinama į kitą būseną su vienodomis tikimybėmis. Taip pat panaudosime matricę iš [Ryd08]:

$$A = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}.$$

Išbandytas paslėptos grandinės ilgis  $N \in \{100, 1000, 10000\}$ . Nustatyti  $\mu_i = i - 1$ , o  $\sigma_1 = \sigma_2 = \dots = \sigma_M \in \{0.1, 0.25, 0.5\}$ .

Gibbs algoritmu gaunamų imčių kokybė priklauso nuo tiriamų duomenų, taigi *burn-in* ir praretinimo parametrai kiekvienam atvejui parinkti individualiai. Tarkim norime gauti  $n = 4000$  nepriklausomų imčių. Pirmą paskaičiuojamos aštuonios paralelios Gibbs grandinės po  $n_{pre} = 2000$  žingsnių. Kiekvieno parametro reikšmių sekai paskaičiuojamos autokoreliacijos ir tai panaudojama atitinkamiems efektyviems imties dydžiams  $ESS_{\theta_i}$  gauti. Nustatomas praretinimo koeficientas  $m = \lceil \frac{n_{pre}}{\min_i ESS_{\theta_i}} \rceil$ . *Burn-in* ilgiui  $B$  nustatyti grandinė suskirstoma į blokus po 100 žingsnių. Tada kiekvienam parametrai ir blokui paskaičiuojamas paskaičiuojamas mažiausias panašumas tarp konkrečios grandinės ir bendro grandinių skirstinio bloke. Iš pradžių jis būna mažas, o kai grandinės priartėja prie stacionaraus skirstinio jos supanašėja. Kiekvienam parametrai kaip *burn-in* ilgis nustatomas žingsnių skaičius kada minimalus panašumas viršijo savo medianą (per visus blokus), o kaip grandinės  $B$  paimamas tų ilgių maksimumas.

Tada paleidžiama grandinė su  $B + mn$  žingsnių, kurių  $B$  yra atmetama ir išrenkamas tik kas  $m$ -asis. Gaunama  $n$  dydžio imtis ir jos sekos autokoreliacija įprastai būna labai maža.

Įverčio palyginimui su tikrąja reikšme naudosime didžiausią skirtumą nuo tikrosios reikšmės  $MD_x = \max_i |x_i - \hat{x}_i|$ , o imties „pločiui“ įvertinti naudosime  $W_x = U_x - L_x$ .  $\sigma$  atveju naudosime santykinę reikšmę,  $MD_\sigma/\sigma$  ir  $W_\sigma/\sigma$ .

Apibendrintų rezultatų lentelė pateikta priede Nr. 1. Kiekvienai parametrai kombinacijai algoritmas buvo paleistas keturis kartus su skirtingomis pradinėmis

pseudoatsitiktinių skaičių generatoriaus būsenom. Lentelėje pateikti vidurkiai ir standartinės paklaidos. 1 pav. grafiškai pavaizduoti rezultatai parametru  $\mu$ .

Galima pamatyti, kad su  $N = 100$  ir didesniu  $\sigma_1$  dažnai gaunami blogi rezultatai, tačiau didinant stebėto proceso ilgį jie pagerėja, kaip ir reikėtų tikėtis.

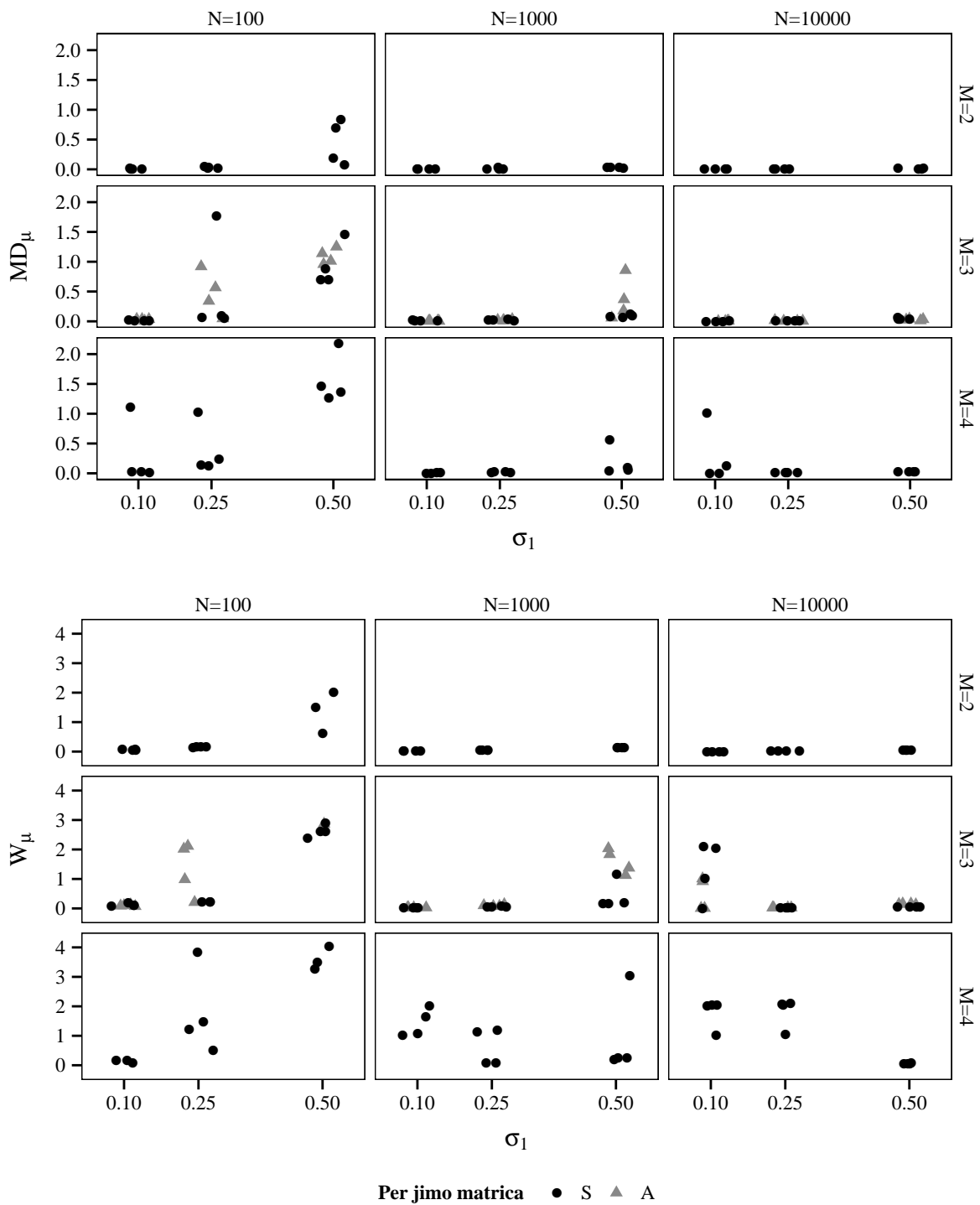
Kadangi apriorinis skirstinys yra invariantiškas būsenų permutacijoms, tai su didesnėmis  $\sigma_i$  reikšmėmis gali būti, kad simuliacijos eigoje būsenos susikeis ir tai pablogins rezultatų kokybę. Tokio atvejo pavyzdys pavaizduotas 2 pav. Vienas iš galimų sprendimų tam yra apriboti vidurkių generavimą, kad visada galėtų  $\mu_1 \leq \mu_2 \leq \dots \leq \mu_M$ , tačiau tai pakeičia aposteriorinio skirstinio formą ir gali sukelti skaičiavimo problemų, kai pora vidurkių yra labai panašūs. Šiuo atveju jau baigus simuliuoti kiekvieną grandinę būsenos buvo pernumeruotos taip, kad vidurkių įverčiai būtų surikiuoti didėjimo tvarka.

Kai kuriais atvejais neįprasti rezultatai gaunami su daugiau skirtingų būsenų. Tada Gibbs algoritmo būsenų erdvė būna daug didesnė ir išnagrinėti visus variantus gali prireikti daugiau žingsnių. Gaunami skirstiniai turi didesnę dispersiją, tačiau vistiek galima gauti gerus įverčius. Baum-Welch kartais neatpažįsta kelių būsenų. Tokio atvejo pavyzdys pateiktas 3 pav.

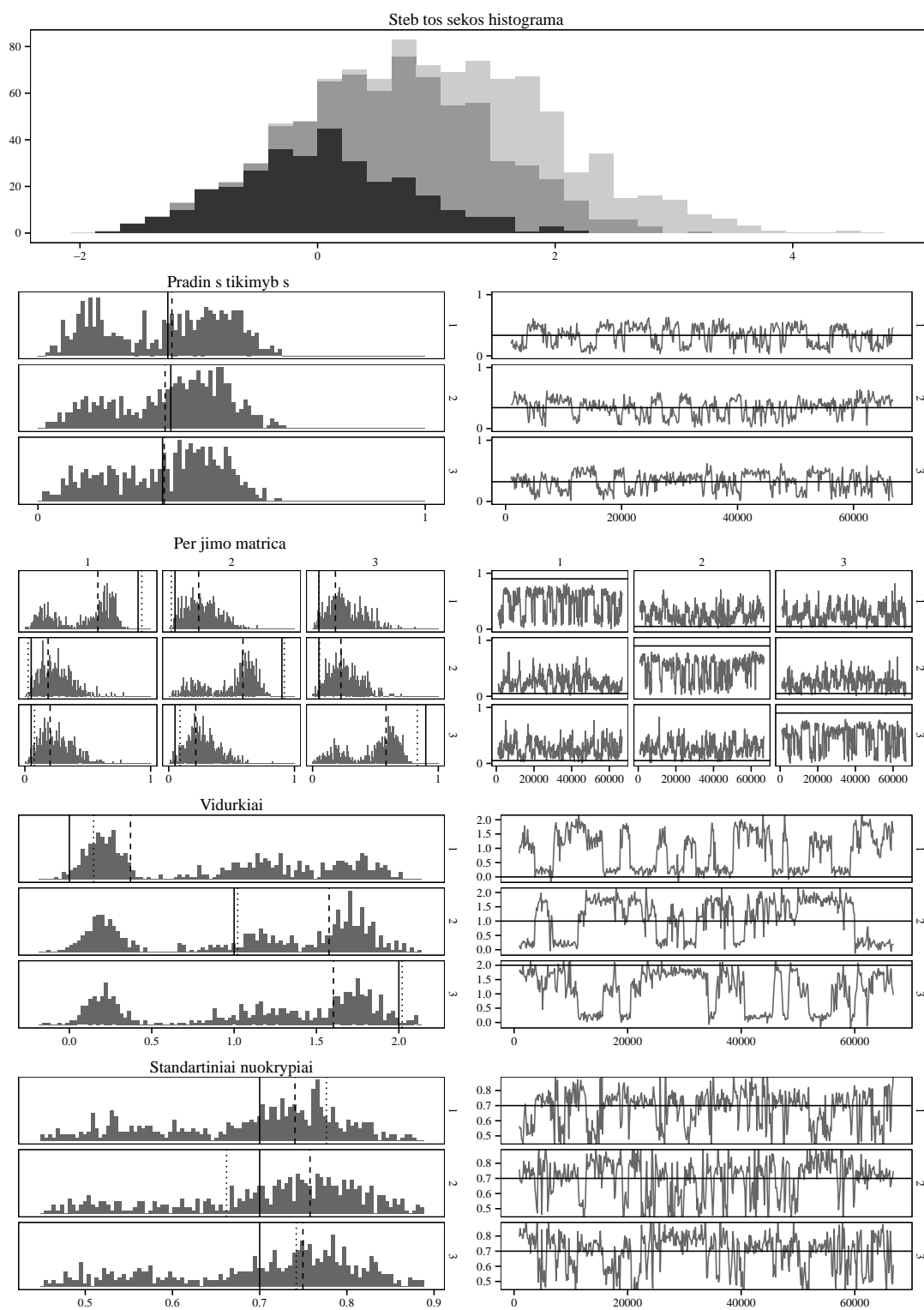
4 pav. pavaizduoti skirstiniai ir įverčiai gauti su skirtingomis dispersijomis  $\sigma = (0.3, 0.5, 0.4)$ ,  $N = 1000$  ir  $T = S_3$ .



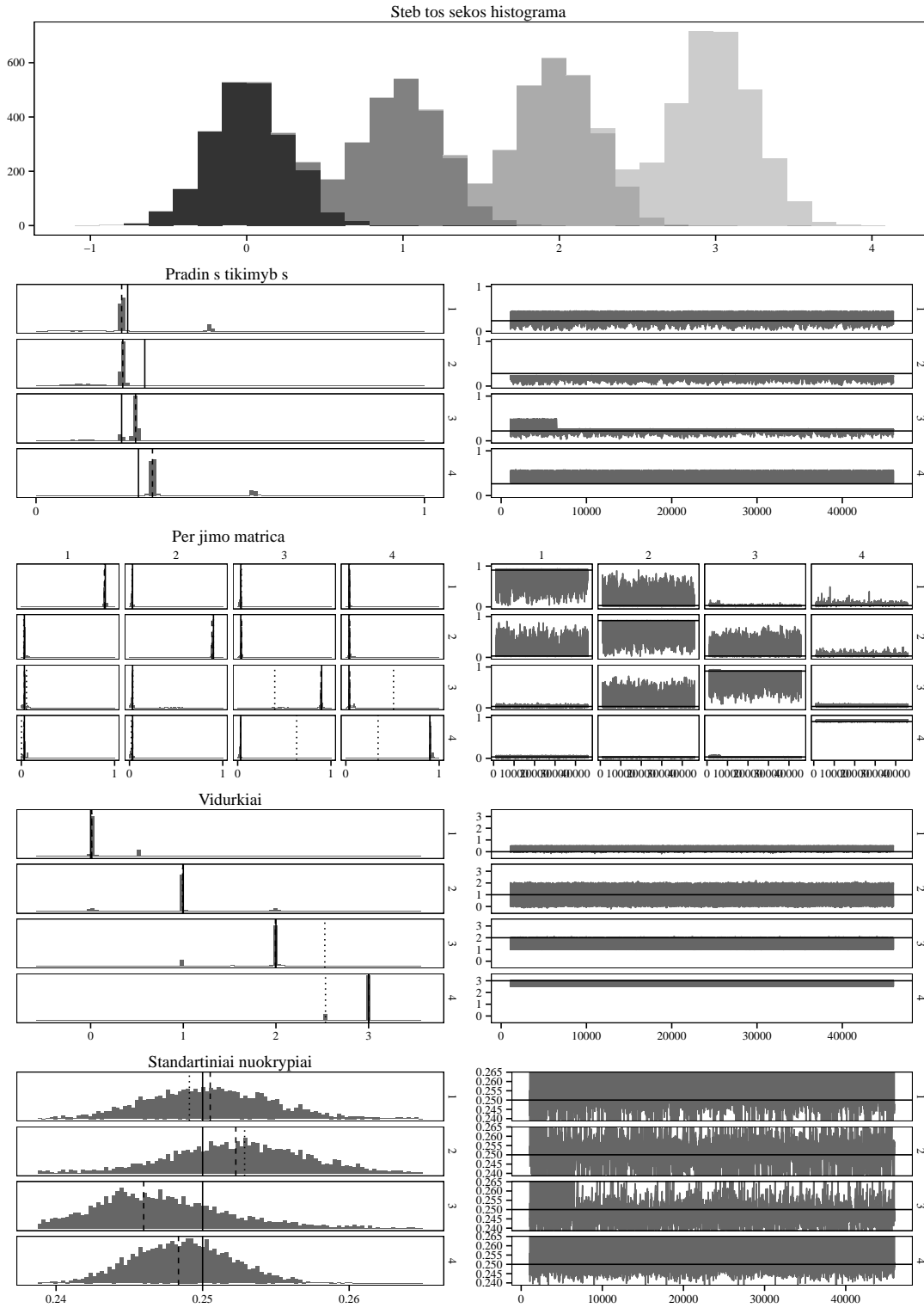
1 pav. Apibendrinti vidurkių įverčių tikslumai ir intervalų pločiai. X ašyje vertės yra nežymiai atsitiktinai pastumdytos, kad geriau matytųsi atskiri taškai.



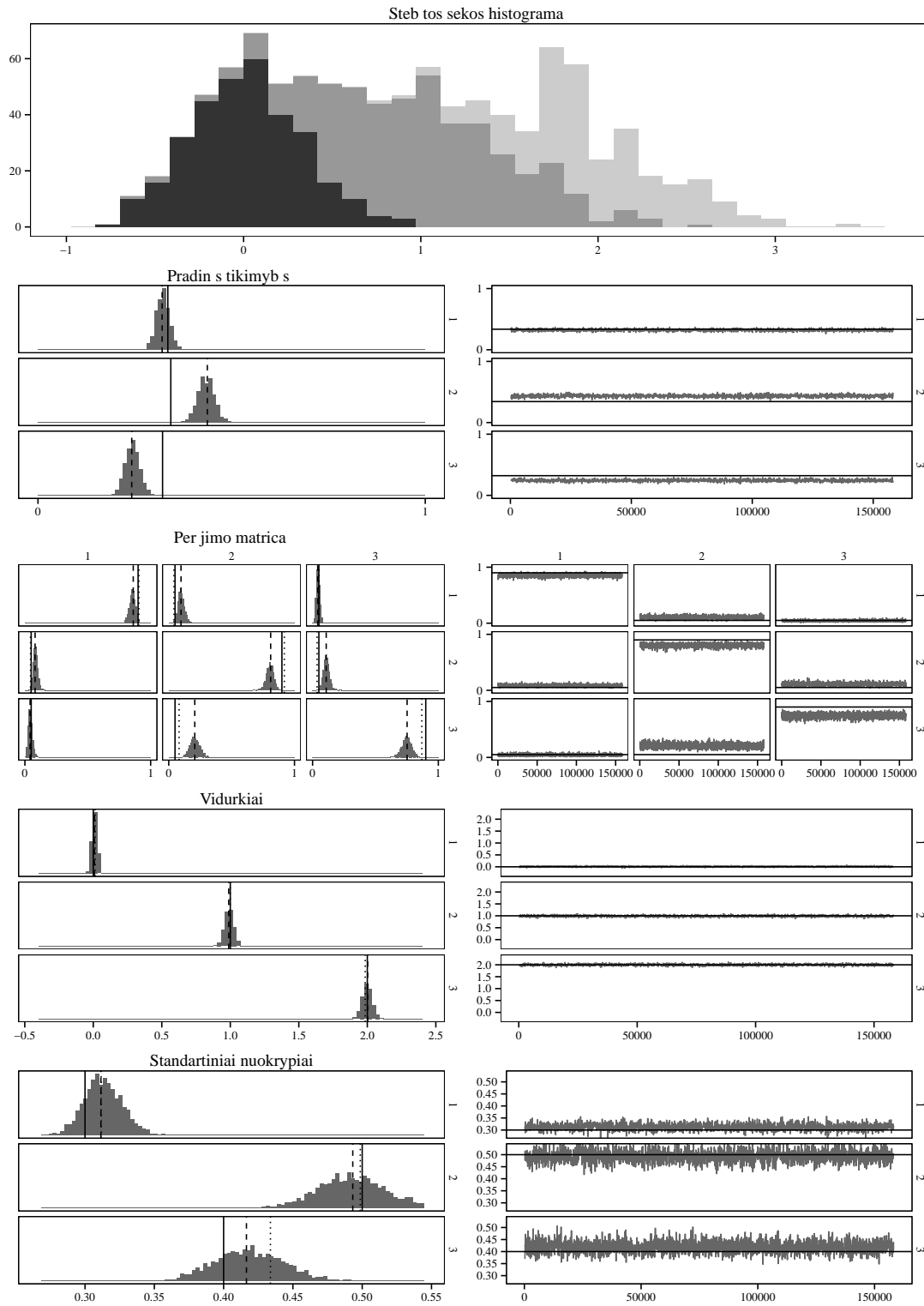
2 pav. Rezultatai su  $N = 1000$ ,  $\sigma = (0.7, 0.7, 0.7)$  ir  $T = S_3$ . Problematiškas atvejis: dėl didelės dispersijos Gibbs algoritmo eigoje būsenos „apsikeičia“ vidurkiais. Histogramose ištisinė linija reiškia tikrąją vertę, brūkšniuota – gautą Gibbs algoritmu, o taškuota – gautą Baum-Welch algoritmu. Dešinėje pavaizduota, kaip keitėsi būseną Gibbs grandinėje.



3 pav. Rezultatai su  $N = 10000$ ,  $\sigma = (0.1, 0.1, 0.1, 0.1)$  ir  $T = S_4$ . Gibbs grandinės būsenų dispersija didelė, tačiau vistiek gaunami teisingi rezultatai. Baum-Welch algoritmas neteisingai nustato porą būsenų.



4 pav. Rezultatai su  $N = 1000$ ,  $\sigma = (0.3, 0.5, 0.4)$  ir  $T = S_3$ .



## Išvados

Modelis mišiniams, aprašytas [RG97], pritaikytas tam atvejui, kai priklausomybę populiacijai nustato nestebima Markovo grandinė.

Sukurta programa, kuri efektyviai įgyvendina modelį, o taip pat pagalbinės funkcijos jos naudojimui su R aplinka. Yra galimybė paraleliai skaičiuoti daug grandinių.

Gaunami geri rezultatai su įvairiomis parametrų kombinacijomis. Problemų iškyla, kai populiacijų vidurkiai būna arti, lyginant su dispersijomis, tačiau atlikus daugiau stebėjimu tai įprastai išsisprendžia. Kaikuriais atvejais pasitaiko šokinėjimas tarp būsenų dėl apriorinio skirstinio invariantiškumo būsenų numerių permutacijoms, o tada net su ilga grandine algoritmas grąžina neteisingus skirstinius. Tai galima būtų išspręsti pridedant papildomų apribojimų būsenų aibei.

Lyginant su Baum-Welch algoritmu paprastesniais atvejais gaunami panašūs rezultatai. Atvejais kai yra daugiau paslėptų būsenų (pavyzdžiui,  $M = 4$ ) Baum-Welch algoritmui kartais nepavyksta jų visų atpažinti ir jį reikia vykdyti daug kartų su skirtingomis pradinėmis reikšmėmis, norint gauti pilnesnį vaizdą. Tuo tarpu tirtu Bajesiniu metodu gauti skirstiniai tais atvejais turi didelę dispersiją, tačiau vistiek teisingai nustato parametrų vertes.



## Summary

### Bayesian parameter estimation for hidden Markov models

*Kiprianas Spiridonovas*

A normal mixture model where a value's population depends on the unobservable state of a Markov chain is a hidden Markov model. Given an observed sequence of values, the problem of estimating the parameters of the underlying process (transition probabilities, means and variances) is usually solved using the Baum-Welch algorithm, which gives point estimates that are local maximums of the likelihood function. We explore an alternative method, which uses the Gibbs sampler with a hierarchical Bayesian model to obtain posterior distributions for the parameters based on weak priors. We adapt a similar method used for general mixture models for the hidden Markov model case. The obtained distributions can be used for performance assessment and inference.





## Literatūros sąrašas

- [ABH72] D. Andrews, P. Bickel, ir F. Hampel. Robust estimates of location: survey and advances. *Journal of the American Statistical Association*, 1972. 69 (345), p. 272–274.
- [BPSW70] L. Baum, T. Petrie, G. Soules, ir N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 1970. 41 (1), p. 164–171.
- [CS99] M. Chen ir Q. Shao. Monte Carlo estimation of Bayesian credible and HPD intervals. *Journal of Computational and Graphical Statistics*, 1999, p. 1–27.
- [FS14] S. Frohwrth-Schnatter. Markov Chain Monte Carlo Estimation of Classical and Dynamic Switching and Mixture Models. *Journal of the American Statistical Association*, 2014. 96 (453), p. 194–209.
- [GG84] S. Geman ir D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence*, 1984.
- [Has70] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 1970. 57 (1), p. 97–109.
- [MRR<sup>+</sup>53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, ir E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953. 21 (6), p. 1087–1092.
- [RG97] S. Richardson ir P. J. Green. On Bayesian analysis of mixtures with an unknown number of components-Discussion. *Journal of the Royal Statistical Society*, 1997. 59 (4), p. 731–792.
- [Ryd08] T. Rydén. EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective. *Bayesian Analysis*, 2008. 3 (4), p. 659–688.
- [Tie94] L. Tierney. Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, 1994. 22 (4), p. 1701–1728.



# Priedas Nr. 1

## Lentelė su apibendrintais testų rezultatais

| $T$   | $N$   | $\sigma_1$ | $\max_i \rho_i$ | $MD_\mu$    | $W_\mu$      | $MD_\sigma/\sigma$ | $W_\sigma/\sigma$ | $MD_T$      | $MD_p$      |
|-------|-------|------------|-----------------|-------------|--------------|--------------------|-------------------|-------------|-------------|
| A     | 100   | 0.10       | 0.182           | 0.028       | 0.097        | 0.253              | 0.657             | 0.271       | 0.037       |
|       |       |            | $\pm 0.012$     | $\pm 0.015$ | $\pm 0.023$  | $\pm 0.030$        | $\pm 0.140$       | $\pm 0.072$ | $\pm 0.033$ |
| A     | 100   | 0.25       | 0.722           | 0.469       | 1.333        | 0.208              | 1.651             | 0.228       | 0.186       |
|       |       |            | $\pm 0.377$     | $\pm 0.368$ | $\pm 0.907$  | $\pm 0.132$        | $\pm 0.721$       | $\pm 0.049$ | $\pm 0.067$ |
| A     | 100   | 0.50       | 0.483           | 1.090       | 31.332       | 0.269              | 1.457             | 0.303       | 0.247       |
|       |       |            | $\pm 0.062$     | $\pm 0.131$ | $\pm 33.625$ | $\pm 0.163$        | $\pm 0.209$       | $\pm 0.073$ | $\pm 0.005$ |
| A     | 1000  | 0.10       | 0.363           | 0.007       | 0.029        | 0.071              | 0.205             | 0.225       | 0.021       |
|       |       |            | $\pm 0.361$     | $\pm 0.002$ | $\pm 0.002$  | $\pm 0.041$        | $\pm 0.015$       | $\pm 0.004$ | $\pm 0.003$ |
| A     | 1000  | 0.25       | 0.416           | 0.020       | 0.098        | 0.105              | 0.284             | 0.241       | 0.028       |
|       |       |            | $\pm 0.350$     | $\pm 0.010$ | $\pm 0.028$  | $\pm 0.053$        | $\pm 0.075$       | $\pm 0.037$ | $\pm 0.014$ |
| A     | 1000  | 0.50       | 0.960           | 0.367       | 1.593        | 0.098              | 0.628             | 0.379       | 0.099       |
|       |       |            | $\pm 0.010$     | $\pm 0.351$ | $\pm 0.416$  | $\pm 0.057$        | $\pm 0.097$       | $\pm 0.034$ | $\pm 0.113$ |
| A     | 10000 | 0.10       | 0.585           | 0.004       | 0.487        | 0.015              | 1.903             | 0.210       | 0.014       |
|       |       |            | $\pm 0.458$     | $\pm 0.004$ | $\pm 0.553$  | $\pm 0.002$        | $\pm 2.124$       | $\pm 0.010$ | $\pm 0.009$ |
| A     | 10000 | 0.25       | 0.562           | 0.009       | 0.026        | 0.040              | 0.081             | 0.224       | 0.014       |
|       |       |            | $\pm 0.462$     | $\pm 0.006$ | $\pm 0.001$  | $\pm 0.014$        | $\pm 0.002$       | $\pm 0.008$ | $\pm 0.007$ |
| A     | 10000 | 0.50       | 0.227           | 0.028       | 0.148        | 0.044              | 0.132             | 0.409       | 0.014       |
|       |       |            | $\pm 0.089$     | $\pm 0.012$ | $\pm 0.014$  | $\pm 0.022$        | $\pm 0.009$       | $\pm 0.011$ | $\pm 0.007$ |
| $S_2$ | 100   | 0.10       | 0.161           | 0.012       | 0.063        | 0.128              | 0.440             | 0.046       | 0.070       |
|       |       |            | $\pm 0.009$     | $\pm 0.004$ | $\pm 0.008$  | $\pm 0.013$        | $\pm 0.062$       | $\pm 0.012$ | $\pm 0.049$ |
| $S_2$ | 100   | 0.25       | 0.161           | 0.029       | 0.160        | 0.120              | 0.462             | 0.041       | 0.073       |
|       |       |            | $\pm 0.011$     | $\pm 0.012$ | $\pm 0.015$  | $\pm 0.107$        | $\pm 0.034$       | $\pm 0.038$ | $\pm 0.074$ |
| $S_2$ | 100   | 0.50       | 0.732           | 0.449       | 20.253       | 0.166              | 1.026             | 0.156       | 0.156       |
|       |       |            | $\pm 0.258$     | $\pm 0.372$ | $\pm 37.744$ | $\pm 0.065$        | $\pm 0.378$       | $\pm 0.128$ | $\pm 0.157$ |
| $S_2$ | 1000  | 0.10       | 0.163           | 0.004       | 0.018        | 0.037              | 0.126             | 0.011       | 0.027       |
|       |       |            | $\pm 0.012$     | $\pm 0.001$ | $\pm 0.001$  | $\pm 0.011$        | $\pm 0.004$       | $\pm 0.004$ | $\pm 0.025$ |
| $S_2$ | 1000  | 0.25       | 0.156           | 0.012       | 0.048        | 0.038              | 0.145             | 0.017       | 0.033       |
|       |       |            | $\pm 0.005$     | $\pm 0.010$ | $\pm 0.002$  | $\pm 0.021$        | $\pm 0.003$       | $\pm 0.004$ | $\pm 0.040$ |
| $S_2$ | 1000  | 0.50       | 0.156           | 0.030       | 0.138        | 0.046              | 0.172             | 0.165       | 0.033       |
|       |       |            | $\pm 0.015$     | $\pm 0.009$ | $\pm 0.002$  | $\pm 0.027$        | $\pm 0.007$       | $\pm 0.013$ | $\pm 0.030$ |
| $S_2$ | 10000 | 0.10       | 0.157           | 0.002       | 0.006        | 0.013              | 0.039             | 0.006       | 0.011       |
|       |       |            | $\pm 0.010$     | $\pm 0.001$ | $\pm 0.000$  | $\pm 0.006$        | $\pm 0.000$       | $\pm 0.003$ | $\pm 0.007$ |
| $S_2$ | 10000 | 0.25       | 0.164           | 0.003       | 0.015        | 0.012              | 0.045             | 0.009       | 0.011       |
|       |       |            | $\pm 0.012$     | $\pm 0.002$ | $\pm 0.000$  | $\pm 0.007$        | $\pm 0.001$       | $\pm 0.003$ | $\pm 0.010$ |
| $S_2$ | 10000 | 0.50       | 0.165           | 0.014       | 0.040        | 0.035              | 0.052             | 0.135       | 0.015       |
|       |       |            | $\pm 0.006$     | $\pm 0.003$ | $\pm 0.001$  | $\pm 0.009$        | $\pm 0.001$       | $\pm 0.011$ | $\pm 0.011$ |
| $S_3$ | 100   | 0.10       | 0.335           | 0.013       | 0.125        | 0.179              | 0.788             | 0.134       | 0.205       |
|       |       |            | $\pm 0.320$     | $\pm 0.003$ | $\pm 0.051$  | $\pm 0.053$        | $\pm 0.267$       | $\pm 0.089$ | $\pm 0.107$ |
| $S_3$ | 100   | 0.25       | 0.324           | 0.494       | 11.311       | 0.118              | 1.049             | 0.111       | 0.158       |
|       |       |            | $\pm 0.325$     | $\pm 0.852$ | $\pm 22.170$ | $\pm 0.070$        | $\pm 0.788$       | $\pm 0.073$ | $\pm 0.063$ |

Pratęsta kitame puslapyje.

| $T$   | $N$   | $\sigma_1$ | $\max_i \rho_i$ | $MD_\mu$    | $W_\mu$      | $MD_\sigma/\sigma$ | $W_\sigma/\sigma$ | $MD_T$      | $MD_p$      |
|-------|-------|------------|-----------------|-------------|--------------|--------------------|-------------------|-------------|-------------|
| $S_3$ | 100   | 0.50       | 0.512           | 0.937       | 2.630        | 0.108              | 1.145             | 0.113       | 0.030       |
|       |       |            | $\pm 0.131$     | $\pm 0.358$ | $\pm 0.215$  | $\pm 0.036$        | $\pm 0.093$       | $\pm 0.088$ | $\pm 0.013$ |
| $S_3$ | 1000  | 0.10       | 0.172           | 0.009       | 0.023        | 0.051              | 0.162             | 0.032       | 0.074       |
|       |       |            | $\pm 0.013$     | $\pm 0.006$ | $\pm 0.001$  | $\pm 0.023$        | $\pm 0.009$       | $\pm 0.005$ | $\pm 0.035$ |
| $S_3$ | 1000  | 0.25       | 0.543           | 0.020       | 0.062        | 0.055              | 0.193             | 0.024       | 0.053       |
|       |       |            | $\pm 0.433$     | $\pm 0.008$ | $\pm 0.002$  | $\pm 0.016$        | $\pm 0.019$       | $\pm 0.011$ | $\pm 0.022$ |
| $S_3$ | 1000  | 0.50       | 0.435           | 0.092       | 0.421        | 0.101              | 0.301             | 0.281       | 0.067       |
|       |       |            | $\pm 0.379$     | $\pm 0.026$ | $\pm 0.494$  | $\pm 0.015$        | $\pm 0.085$       | $\pm 0.070$ | $\pm 0.059$ |
| $S_3$ | 10000 | 0.10       | 0.778           | 0.002       | 1.290        | 0.020              | 3.280             | 0.007       | 0.035       |
|       |       |            | $\pm 0.404$     | $\pm 0.001$ | $\pm 0.992$  | $\pm 0.007$        | $\pm 2.155$       | $\pm 0.001$ | $\pm 0.021$ |
| $S_3$ | 10000 | 0.25       | 0.580           | 0.005       | 0.019        | 0.012              | 0.058             | 0.011       | 0.021       |
|       |       |            | $\pm 0.475$     | $\pm 0.002$ | $\pm 0.002$  | $\pm 0.005$        | $\pm 0.003$       | $\pm 0.003$ | $\pm 0.007$ |
| $S_3$ | 10000 | 0.50       | 0.343           | 0.041       | 0.056        | 0.048              | 0.081             | 0.236       | 0.018       |
|       |       |            | $\pm 0.324$     | $\pm 0.013$ | $\pm 0.003$  | $\pm 0.011$        | $\pm 0.004$       | $\pm 0.015$ | $\pm 0.010$ |
| $S_4$ | 100   | 0.10       | 0.729           | 0.294       | 4.002        | 0.200              | 0.878             | 0.138       | 0.180       |
|       |       |            | $\pm 0.371$     | $\pm 0.541$ | $\pm 7.720$  | $\pm 0.065$        | $\pm 0.205$       | $\pm 0.120$ | $\pm 0.059$ |
| $S_4$ | 100   | 0.25       | 0.767           | 0.380       | 1.759        | 0.237              | 1.558             | 0.163       | 0.192       |
|       |       |            | $\pm 0.376$     | $\pm 0.435$ | $\pm 1.443$  | $\pm 0.045$        | $\pm 0.800$       | $\pm 0.121$ | $\pm 0.063$ |
| $S_4$ | 100   | 0.50       | 0.618           | 1.567       | 12.301       | 0.094              | 1.131             | 0.067       | 0.028       |
|       |       |            | $\pm 0.045$     | $\pm 0.412$ | $\pm 17.402$ | $\pm 0.016$        | $\pm 0.079$       | $\pm 0.022$ | $\pm 0.019$ |
| $S_4$ | 1000  | 0.10       | 0.812           | 0.007       | 1.438        | 0.095              | 5.663             | 0.025       | 0.048       |
|       |       |            | $\pm 0.356$     | $\pm 0.004$ | $\pm 0.482$  | $\pm 0.027$        | $\pm 2.423$       | $\pm 0.010$ | $\pm 0.040$ |
| $S_4$ | 1000  | 0.25       | 0.780           | 0.019       | 0.617        | 0.067              | 0.807             | 0.034       | 0.084       |
|       |       |            | $\pm 0.405$     | $\pm 0.003$ | $\pm 0.626$  | $\pm 0.011$        | $\pm 0.665$       | $\pm 0.015$ | $\pm 0.022$ |
| $S_4$ | 1000  | 0.50       | 0.375           | 0.189       | 0.938        | 0.120              | 0.360             | 0.225       | 0.084       |
|       |       |            | $\pm 0.411$     | $\pm 0.251$ | $\pm 1.407$  | $\pm 0.066$        | $\pm 0.129$       | $\pm 0.080$ | $\pm 0.021$ |
| $S_4$ | 10000 | 0.10       | 0.770           | 0.283       | 1.778        | 0.017              | 5.614             | 0.015       | 0.029       |
|       |       |            | $\pm 0.088$     | $\pm 0.485$ | $\pm 0.498$  | $\pm 0.003$        | $\pm 2.439$       | $\pm 0.012$ | $\pm 0.004$ |
| $S_4$ | 10000 | 0.25       | 0.992           | 0.009       | 1.816        | 0.020              | 1.431             | 0.012       | 0.033       |
|       |       |            | $\pm 0.002$     | $\pm 0.003$ | $\pm 0.517$  | $\pm 0.004$        | $\pm 0.083$       | $\pm 0.004$ | $\pm 0.017$ |
| $S_4$ | 10000 | 0.50       | 0.303           | 0.029       | 0.061        | 0.066              | 0.088             | 0.227       | 0.036       |
|       |       |            | $\pm 0.253$     | $\pm 0.004$ | $\pm 0.004$  | $\pm 0.013$        | $\pm 0.003$       | $\pm 0.019$ | $\pm 0.011$ |