Segumudeli õppimine osaliselt sildistatud andmetest

Bakalaureusetöö

Tanel Pärnamaa

Matemaatilise statistika instituut

Matemaatika-informaatikateaduskond

Tartu Ülikool

Juhendajad: Leopold Parts, Raivo Kolde

2013

Sisukord

Nomenklatuur

N	treeningandmete arv
N^*	testandmete arv
D	andmete dimensioon
K	komponentide (klastrite) koguarv
C	teadaolevate klastrite arv
$\mathbf{X}$	N x D andmemaatriks, treeningandmed
$\mathbf{X}^*$	N^* x D andmemaatriks, testandmed
π_k	komponendi osakaal
$\boldsymbol{\mu}_k$	komponendi k keskmine
$\boldsymbol{\Sigma}_k$	komponendi k kovariatsioonimaatriks
$\boldsymbol{\theta}_k$	klastri k parameetrid
z_n	latente tunnus, klassikuuluvuse näitaja
$\mathbf{Z}$	N x K maatriks latentsetest tunnustest
α	Dirichlet' jaotuse ja Dirichlet' protsessi kontsentratsiooni parameeter
G_0	baasjaotus
A^c	hulga A täiend
δ_x	Diraci mõõt

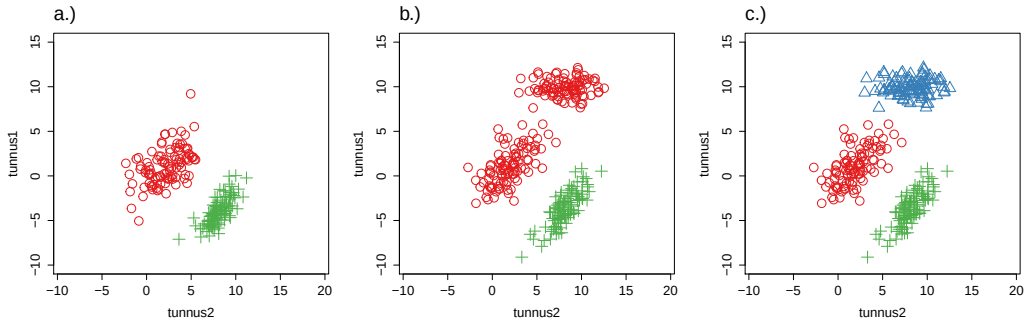
Sissejuhatus

Kuigi tavaelus on diskrimineerimine inetu ja karistatav, siis paljud statistikud töötavad igapäevaselt selle nimel, et üht tüüpi objekte paremini ülejäänutest eristada. Kas tulnud e-mail on spämmikiri või mitte? Kas anda pangalaenu mehele, kes on olnud 2 kuud töötu, aga eelnevalt teenis 1000 eurot kuus? Diskrimineerimine ongi statistika osa, mis tegeleb nende küsimustega. Uuritakse meetodeid, kuidas andmepunkte eristada ja liigitada etteantud klassidesse. Diskrimineerimise sünonüümina kasutatakse sageli leebemalt kõlavat sõna klassifitseerimine.

Klassikalised diskrimineerimismeetodid (näiteks lineaarne diskriminantanalüüs, otsustuspuud, SVM) eeldavad vaikimisi, et uuritava populatsiooni kõik klassid on esindatud treeningandmetes ehk andmetes, mille põhjal klassifitseerimisalgoritm koostatakse. See eeldus on sageli aga liialt range. Näiteks bioloogiliste andmete korral võib uuritav populatsioon kiiresti areneda ja muutuda ning sildistamist vajavates andmetes võib esineda uusi gruppe. Samuti võivad ettevõtte kliendibaas või tarbijate harjumused muutuda ning klassikalised klassifitseerimisalgoritmid ei ole võimelised neid muutusi tuvastama.

Joonisel 1 illustreerime antud probleemi. Nimetame klassifitseerijat hetkeks hellitavalt bioloogiks ning oletame, et ta peab kahe verenäitaja põhjal kindlaks tegema, kas laborihiirtel on hea- või pahaloomuline kasvaja. Bioloogi on õpetatud eristama ainult neid kahte klassi. Seega iga järgneva hiire liigitab bioloog kas hea- või pahaloomulise kasvajaga gruppi, kuigi võiksime andmetest tuvastada, et pahaloomulise kasvajaga hiirte grupp jaguneb tegelikult kaheks, millest ühe osagrupi ravi võiks olla palju kergemini teostatav. Kahe klassi ja kahe muutuja korral on kerge visualiseerida, mis andmetes toimub, aga kui klasse on kümneid ja muutujaid sadu, on keeruline analüüsi sarnaselt kohandada.

Seega tekib küsimus, mida teha juhtudel, kui sildistamist vajavates andmetes



Joonis 1: Klassifitseerija õpitakse treeningandmetelt (joonis a). Klassikalised diskrimineerimisalgoritmid ei ole võimelised tuvastama uusi andmegruppe ja liigitavad uue klassi andmed mõnda teadaolevasse klassi (joonis b). Joonisel (c) on näidatud tegelikud testandmed ja seega tulemus, mida soovime, et meie algoritm tagastaks.

võib esineda uusi gruppe, mida treeningandmetes pole nähtud? Selle töö eesmärk ongi uurida ja implementeerida klassifitseerijaid, mis on treenitud osaliselt sildistatud andmetelt ja on võimelised tuvastama uusi andmegruppe.

Rangemalt, olgu meil treeningandmed $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ ja testandmed $\mathbf{X}^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_{N^*}^*\}$, mis koosnevad D-mõõtmelistest andmepunktidest $\mathbf{x}_n$, $\mathbf{x}_m^* \in \mathbb{R}^D$. Andmed jagunevad C-sse erinevasse klassi, kus C võib olla meile teadmata. Treeningandmete $\mathbf{X}$ puhul teame iga n korral $\mathbf{x}_n$ klassi c_n, kuid testandmete $\mathbf{X}^*$ jaoks see info puudub. Meie eesmärgiks on sobitada funktsioon $g : \mathbb{R}^D \to \{1, ..., C\}$, mis tagastab suvalise andmepunkti jaoks, millisesse klassi see kuulub. Sealjuures soovime, et $\sum_n \delta_{g(x_n)c_n}$ oleks võimalikult suur, s.t. teadaolevate andmete klassifitseerimine oleks võimalikult täpne. Edukaks tulemuseks on funktsioon g, mis on esitatud tema parameetrite $\boldsymbol{\theta}$ kaudu ning kasutatud klasside arv $K \leq C$.

Töö koosneb kolmest osast. Esimeses kirjeldatakse kahte algoritmi osaliselt sildistatud andmete klassifitseerimiseks. Need meetodid põhinevad Gaussi segumudelil ja EM-algoritmil ning sobiv klastrite arv valitakse Bayesi informatsioonikriteeriumi põhjal. Seejärel pöördume mitteparameetrilise Bayesi statistika valdkonda: andes Bayesi segumudeli korral komponentide osakaalude eeljaotuseks Dirichlet protsessi, järeldab mudel vajalike klastrite arvu automaatselt ja pääseme subjektiivsest mudeli valikust. Seda mudelit kutsume Dirichlet protsessi segu-

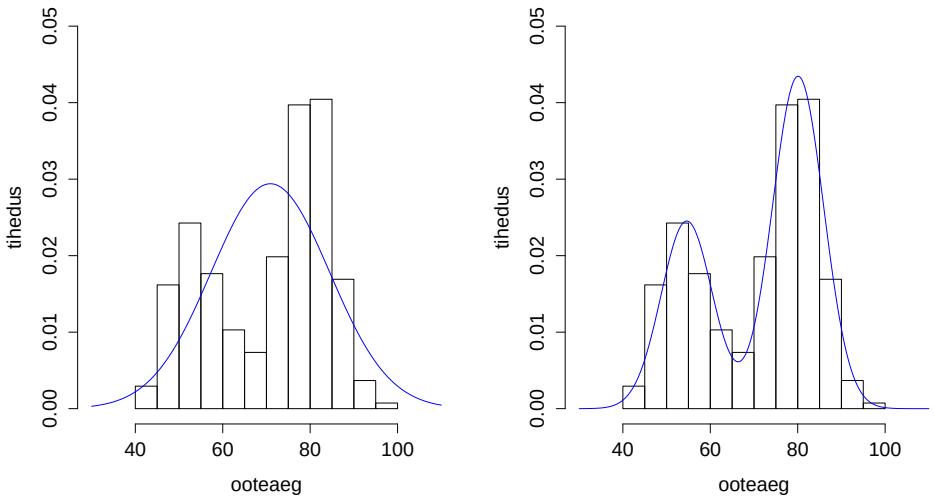
mudeliks. Viimases osas on algoritmide headust testitud nii genereeritud kui ka reaalsetel andmestikel. Kõik kirjeldatud mudelid on implementeeritud ja joonised on koostatud statistikatarkvaras R.

Soovin tänada oma juhendajaid: Leopold Partsi ja Raivo Koldet. Erilised tänusõnad kuuluvad Leopoldile tema huvitava teemapüstituse, lõbusa suhtumise, põnevate selgituste, põhjaliku tagasiside, kannatlikkuse ja arvukate paranduste eest.

Peatükk 1

Gaussi segumudel

Reaalsete andmestike modelleerimiseks ei piisa ühekomponendilisest normaaljaotusest. Vaatleme näidet, mis on toodud joonisel 1.1. Andmete jaotus on selgelt bimodaalne ning ühekomponendiline normaaljaotus ei ole sobilik selle modelleerimiseks. Kahe normaaljaotuse lineaarkombinatsioon kirjeldab andmete struktuuri paremini.



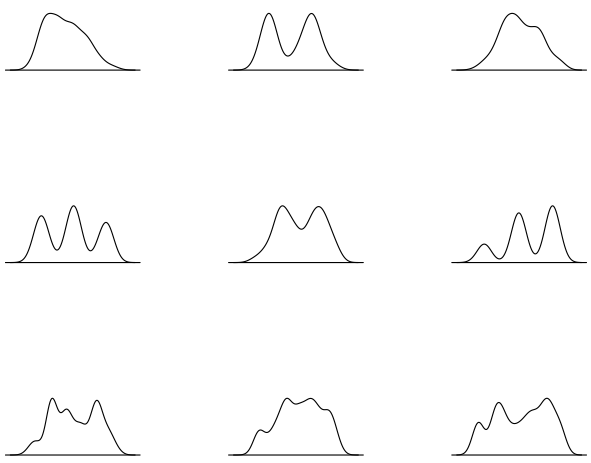
Joonis 1.1: Illustratsioon ühekomponendilise normaaljaotuse puudulikkusest, kui on proovitud modelleerida kuumaveeallika *Old Faithful* pursetevahelist aega.

1.1 Segumudeli kirjeldus

Segujaotus on jaotus, mis on formuleeritud lihtsamate baasjaotuste (näiteks normaaljaotuste) lineaarkombinatsioonina. Kasutades piisaval arvul baasjaotuseid ja sättides iga komponendi parameetreid on segumudeli abil võimalik kõiki jaotuseid suvalise täpsuseni lähendada. Erinevaid näiteid kahe-, kolme- ja viiekomponendilistest segujaotustest on toodud joonisel 1.2. Segujaotus kirjutatakse üldkujul:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}|\boldsymbol{\theta}_k) \ , \tag{1.1}$$

kus $\mathbf{x}$ on D-mõõtmeline andmevektor, mis on kirjeldatud K baasjaotuse $f_k(\mathbf{x}|\boldsymbol{\theta}_k)$ abil, kus komponendi k kaal on π_k, baasjaotuse f_k parameetrid on θ_k ning $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$.



Joonis 1.2: Näited Gaussi segujaotustest $K = 2, 3, 5$ korral (ridade kaupa).

Kui segujaotus moodustatakse K normaaljaotusest, kutsutakse seda Gaussi segumudeliks ning see kirjutatakse kujul:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \ , \tag{1.2}$$

kus

$$f_k(\mathbf{x}|\boldsymbol{\theta}_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\} . \tag{1.3}$$

Andmete klassifitseerimisel või klasterdamisel soovime kindlaks teha segumudeli komponendi, mis genereeris antud andmepunkti. Seega on mõistlik sisse tuua latentse (peidetud) tunnuse mõiste. Me nimetame latentseks tunnuseks $\mathbf{z}$ binaarset K-mõõtmelist vektorit, kus leidub indeks k selliselt, et $z_k = 1$. See latentne tunnus iseloomustab andmepunkti kuuluvust teatud segujaotuse komponenti: näiteks $\mathbf{z} = (0, 0, 1, 0)$ tähistab, et andmepunkt kuulub klassi 3. Kasutame ka tähistust z_{nk}, mis kirjeldab andmepunkti $\mathbf{x}_n$ latentse tunnuse k-ndat komponenti.

Andmepunkti $\mathbf{x}$ ja tema latentse tunnuse $\mathbf{z}$ ühisjatuse $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ saame kirjutada:

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z}|\boldsymbol{\theta})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} . \tag{1.4}$$

Me kutsume $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ täisjaotuseks. Võime mõelda, et nähtud andmed on genereeritud selliselt, et kõigepealt valitakse andmepunktile klass k proportsionaalselt $\boldsymbol{\pi}$ väärtustele, seejärel tõmmatakse $\mathbf{x}_n$ väärtused vastavalt parameetritele $\boldsymbol{\theta}_k$. Meid huvitab, milline generatiivne protsess kirjeldab andmeid kõige paremini.

1.2 Parameetrite hindamine

Järgnevalt kirjeldame, kuidas segujaotuse korral mudeli parameetreid hinnata. Antud juhul on üldiselt tegemist mittekumera funktsiooni optimeerimisega. See tähendab, et tõepära lokaalseid maksimume võib olla rohkem kui üks ning üldiselt ei eksisteeri kinnises vormis analüütilisi lahendusi nagu ühekomponendiliste jaotuste parameetrite hindamisel kasutatakse.

1.2.1 Probleemi kirjeldus

Olgu meil NxD andmemaatriks $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ ja oletame, et on mõistlik arvata, et segumudel K normaaljaotuse komponendiga on meie andmetele sobilik.

Sel juhul on Gaussi segumudel määratud parameetritega $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_K\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_K\}$ ja $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, ..., \boldsymbol{\Sigma}_K\}$. Kuidas hinnata vajalikud parameetrid? Statistiku esimene mõte võiks seostuda suurima tõepära meetodiga.

Normaaljaotuse segumudeli tõepärafunktsiooniks saame (valem 1.2):

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} . \tag{1.5}$$

Kuna logaritmilise tõepäraga on kergem töötada ja selle maksimiseerimine on ekvivalentne tavalise tõepära maksimiseerimisega (sest logaritm on monotoonne funktsioon), siis edasises kasutamegi just logaritmilist tõepära, mis avaldub kujul:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} . \tag{1.6}$$

Paneme tähele, et kui võtta $\mu_1 = \mathbf{x}_1, \boldsymbol{\Sigma}_1 = \sigma_1^2 \mathbf{I}$ ja lastes $\sigma_1 \to 0$, saame logaritmilise tõepära viia ükskõik kui suureks. Segumudeli implementeerimisel tuleb seda arvesse võtta.

Segujaotuse tõepära maksimiseerimine on palju keerukam kui ühekomponendilise normaaljaotuse korral, sest nüüd on tegemist logaritmiga summast. Selle tulemusena ei ole eelneval avaldisel ilusat analüütilist lahendust. Üheks võimaluseks tõepära maksimiseerida on kasutada iteratiivseid meetodeid, nagu näiteks EM-algoritm, mida vaatamegi lähemalt järgmises sektsioonis.

1.2.2 Heuristiline lähenemine EM-algoritmile

Logaritmilist tõepära saab lihtsustada, kui eeldame, et iga andmepunkti korral on teada tema latentne tunnus. Uuritavate andmepunktide latentsete tunnuste hulka tähistame $\mathbf{Z}$. Kuna $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = p(\mathbf{Z}|\boldsymbol{\theta})p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta})$, siis saame logaritmilise tõepära esitada kujul (Bishop [2006]):

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} . \tag{1.7}$$

Nüüd on segujaotuse parameetrite leidmine sarnane ühekomponendilise normaaljaotuse parameetrite leidmisele. Probleem on aga selles, et me tegelikult ei

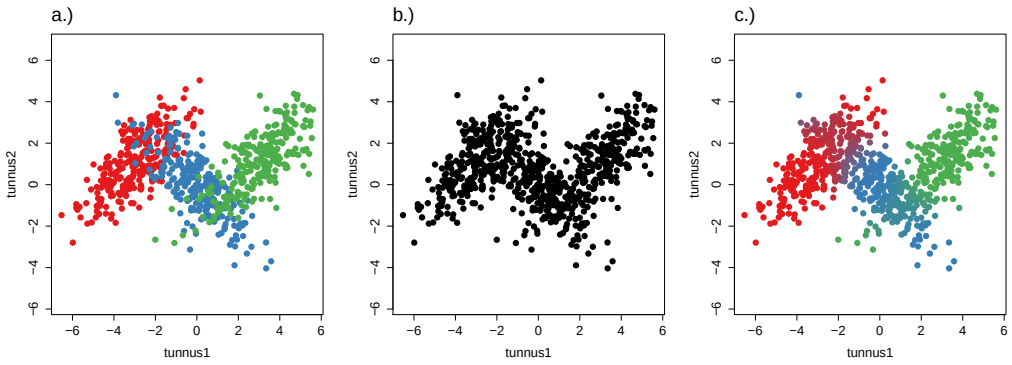
tea z_{nk} väärtuseid. Mis oleks kui kasutaksime z_{nk} asemel tema ooteväärtust?

$$\gamma(z_{nk}) := E(z_{nk}|\mathbf{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \quad p(z_{nk} = 1|\mathbf{x}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) =$$

$$= \frac{p(z_{nk} = 1)p(\mathbf{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\mathbf{x}_n|z_{nj} = 1)} = \qquad \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{1.8}$$

Arvutades $\gamma(z_{nk})$ väärtused ning kasutades neid klastri parameetrite hindamiseks, on garanteeritud, et logaritmiline tõepära ei vähene.

Väärtusest $\gamma(z_{nk})$ võime mõelda kui suurusest, mis näitab, kui suure osa *vastutusest* võtab segujaotuse k-s komponent andmepunkti $\mathbf{x}_n$ kirjeldamisel. Seda oleme illustreerinud joonisel 1.3. Näidatud on nii sildistatud ja sildistamata andmestik kui ka andmepunktide $\gamma(z_{nk})$ väärtused.



Joonis 1.3: $\gamma(z_{nk})$ tähenduse illustreerimine. Näidatud on sildistatud andmestik (a), sildistamata andmestik (b) ja iga andmepunkti jaoks arvutatud $\gamma(z_{nk})$ väärtus (c). Klastrisse kuuluvust kodeerib värv. Andmete kuju on inspireeritud Bishop [2006], kasutatud algoritm on autori tehtud.

EM-algoritmi pseudokood Gaussi segumudeli jaoks on toodud algoritmis 1 (Bishop [2006]). Esiteks on vaja algväärtustada parameetrid. Keskmistele $\boldsymbol{\mu}_k$ võib anda suvalise andmepunkti väärtuse (iga komponendi keskmisele erineva), kõik kovariatsioonimaatriksid võib väärtustada näiteks kogu andmete kovariatsioonimaaktriksiga ning osakaalude komponentidele võib anda võrdse osakaalu. Järgneb E-samm, kus hinnatakse $\gamma(z_{nk})$ väärtused. Seejärel maksimiseeritakse M-sammul andmete tõepära klastri parameetreid uuesti hinnates. E- ja M-sammu korratakse

senikaua kuni logaritmiline tõepära enam ei muutu või etteantud iteratsioonide arv saab täis.

Pseudokood 1 EM algoritm Gaussi segumudeli jaoks

1. Algväärtusta parameetrid $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k ja hinda log-tõepära (valem 1.6).
2. E-samm: arvuta $\gamma(z_{nk})$ väärtused (valem 1.8).
3. M-samm: uuenda komponentide parameetrid, kasutades E-sammu tulemusi

$$\boldsymbol{\mu}_k^{uus} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{uus} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^{uus})(\mathbf{x}_n - \boldsymbol{\mu}_k^{uus})^T$$

$$\pi_k^{uus} = \frac{N_k}{N}$$

kus

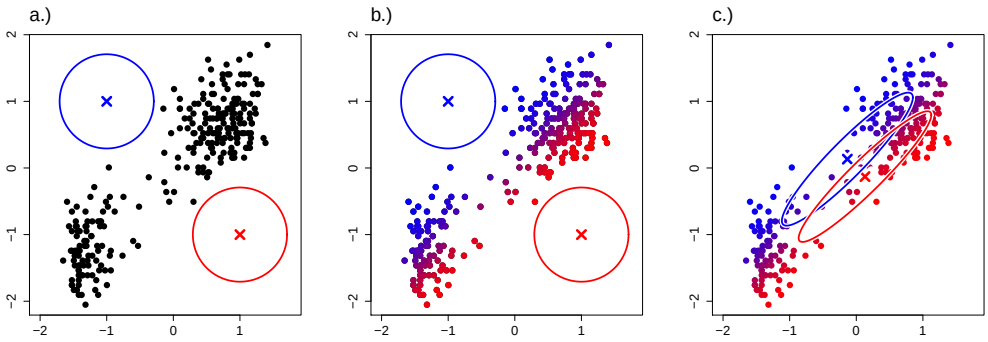
$$N_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})$$

4. Hinda log-tõepära (valem 1.6) ja kontrolli log-tõepära koonduvust. Kui tõepära koonduvuse kriteerium pole täidetud, mine sammu 2 juurde tagasi.

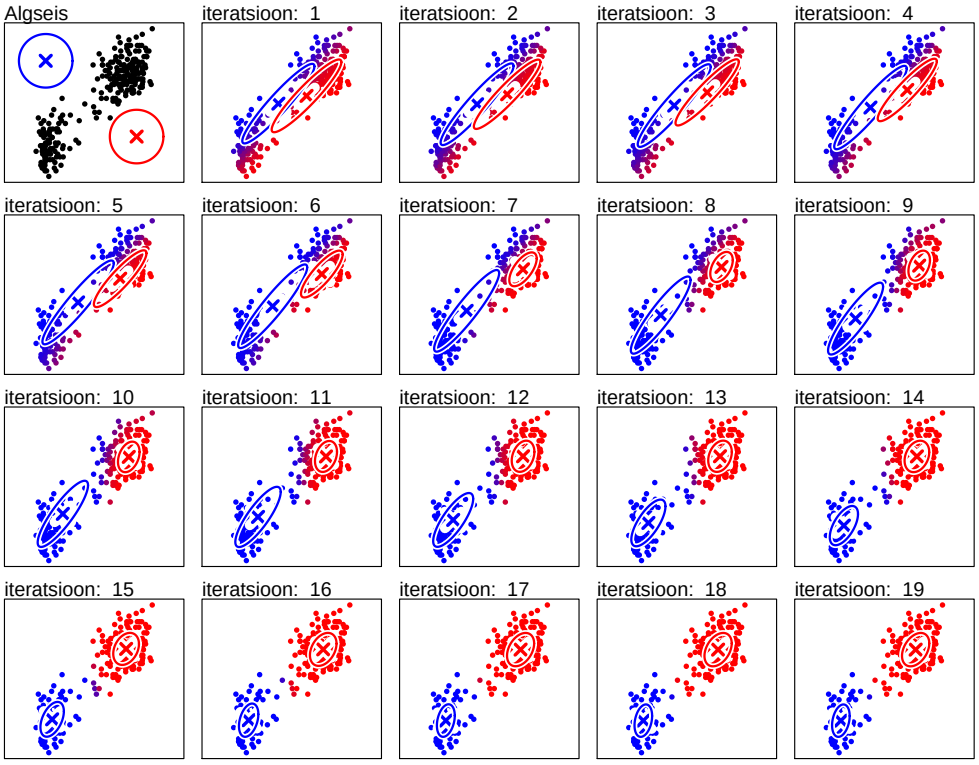
Joonisel 1.4 on näidatud EM-alogiritmi esimene iteratsioon, mis illustreerib E-sammu ja M-sammu tähendust. Joonisel 1.5 on näidatud EM-algoritmi 19 iteratsiooni illustreerimaks algoritmi koondumist. Kasutatud on reaalset andmestikku *Old Faithful*, kus tunnusteks on kuumaveeallika *Old Faithful* purske kestvus minutites ja aeg järgmise purskeni minutites.

1.2.3 EM-algoritmist üldiselt

EM-algoritm on iteratiivne meetod suurima tõepära hinnangu leidmiseks tõenäosuslike mudelite korral. Seda kasutatakse siis, kui otsene tõepära maksimiseerimine on raske, kuid tuues sisse latentsed tunnused muutub tõepära hindamine lihtsamaks. EM-algoritmi on kahesammuline (E-samm ja M-samm) protseduur. Kõigepealt hindame latentsed tunnused ja seejärel leiame parameetrid $\boldsymbol{\theta}$, mis



Joonis 1.4: EM algoritmi esimese iteratsioon. (a) Parameetritele antakse algväärtused. (b) E-samm: arvutatakse $\gamma(z_{nk})$ väärtused, mida on tähistatud värvi abil. (c) M-samm: arvutatakse klastrite uued parameetrid.



Joonis 1.5: EM algoritmi 19 iteratsiooni. Näeme, et algoritm koondub 15 iteratsiooni jooksul.

maksimiseerivad tõepära. Nüüd, kui meil on olemas $\boldsymbol{\theta}$ hinnang, saame leida parema hinnangu latentsetele tunnustele. Seejärel arvutame jälle hinnangu parameetritele $\boldsymbol{\theta}$ ja kordame neid samme kuni algoritm koodub. Selle saame kokku võtta algortimiga Bishop [2006] .

Pseudokood 2 Üldine EM algoritm

1. Algväärtusta parameetrid $\boldsymbol{\theta}$.
2. E-samm: hinda $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(m)})$
3. M-samm: leia parameetrid $\boldsymbol{\theta}^{m+1}$, mis maksimiseerivad:

$$E_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(m)}} \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(m)}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

4. korda samme 2 ja 3 senikaua, kuni algoritm koondub

EM-algoritm ei vähenda kunagi log-tõepära Gupta and Chen [2010]. Tavaliselt leiab EM-algoritm log-tõepära funktsiooni mõne ekstreemumi, kuid pole garanteeritud, et tegu on globaalse ekstreemumiga. Sellepärsat on vajalik EM-algoritmi jooksutada mitu korda erinevate algväärtustega ja lõplikuks $\boldsymbol{\theta}$ hinnanguks valida väärtused, mis saadi katsel, mille tõepära oli suurim. Praktikas võib mudelit jooksutada senikaua, kuni näiteks 5 järjestikuse juhusliku initsialiseerimisega pole parim log-tõepära muutunud. Selliselt toimisin ka mina EM-algoritmil põhinevate mudelite implementeerimisel.

1.3 Segumudel osaliselt sildistatud andmete jaoks

Olgu meil sildistatud treeningandmed $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, mille korral on teada z_{nk} väärtused, ja meie ülesandeks sildistada testandmed $\mathbf{X}^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_{N^*}^*\}$, mille puhul z_{nk}^* on teadmata. Selleks võime treeningandmete põhjal õppida segumudeli ning saadud parameetrite põhjal sildistada testandmed. Mida teha aga siis, kui testandmetes võib olla andmegruppe, mida treeningandmetes ei leidu? Järgmisena ongi kirjeldatud kahte viisi, kuidas hinnata segumudeli parameetrid osaliselt sildistatud andmetelt, et oleksime võimelised tuvastama ka uusi klasse andmetest.

1.3.1 Transduktiivne ja induktiivne mudel

Kui eeldame, et treening- ja testandmed on pärit samast populatsioonist, siis võime kasutada mõlemat valimit mudeli parameetrite hindamiseks. Sellist mudeli parameetrite hindamise viisi nimetame transduktiivseks mudeliks. Tähistades teadaolevate klasside arvu C-ga ja kogu klasside arvu K-ga, saame logaritmiline tõepära kirjutada kujul Bouveyron [2010]:

$$\begin{aligned} \ln p(\mathbf{X}, \mathbf{X}^*, \mathbf{Z}, \mathbf{Z}^* | \boldsymbol{\theta}) = \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) + \ln p(\mathbf{X}^*, \mathbf{Z}^* | \boldsymbol{\theta}) = \\ = \sum_{n=1}^{N} \sum_{k=1}^{C} z_{nk} \ln(\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) + \sum_{n=1}^{N^*} \sum_{k=1}^{K} z^*_{nk} \ln(\pi_k \mathcal{N}(\mathbf{x}^*_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \ . \end{aligned} \tag{1.9}$$

Treeningandmete korral teame klastritesse kuuluvust iseloomustavate suuruste z_{nk} väärtuseid, testandmete korral tuleb need hinnata sarnaselt Gaussi segumudeli parameetrite hindamisega. Algoritm, mis eelnevalt toodud tõepära maksimiseerib, on näidatud pseudokoodis 3 (Bouveyron [2010]).

Transduktiivse mudeli korral hindame teadaolevate klastrite parameetrid igal klassifitseerimiskorral uuesti. See tähendab, et alati kui soovime mudelit kasutada, peab meil ligipääs olema treeningandmetele. Treeningandmed võivad olla mahukad ning mugavam oleks hoida alles ainult mudeli parameetrid kui kõiki treeningandmete andmepunkte. Selline viis oleks mõistlikum ka suurte andmestike klassifitseerimise jaoks. Induktiivseks mudeliks kutsume mudelit, kus teadaolevate klastrite parameetrid on hinnatud treeningandmete põhjal ning neid järgnevalt enam ei muudeta. Logaritmiline tõepära arvutamisel kasutame vaid testandmeid (Bouveyron [2010]):

$$\ln p(\mathbf{X}^*, \mathbf{Z}^* | \boldsymbol{\theta}) = \sum_{n=1}^{N^*} \sum_{k=1}^{K} z^*_{nk} \ln(\pi_k \mathcal{N}(\mathbf{x}^*_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \ . \tag{1.10}$$

Uute klastrite parameetrite hindamine käib sarnaselt klassikalise Gaussi segumudeli parameetrite hindamisega. Ainuke erinevus tuleb sisse komponentide osakaalude hindamisega. Täpne algoritm on näidatud pseudokoodis 4.

Pseudokood 3 Transduktiivne mudel

1. Algväärtusta $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k ja hinda log-tõepära (valem 1.9).
2. E-samm: treeningandmete $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ korral $\gamma(z_{nk}) = z_{nk}$, testandmete $\{\mathbf{x}_1^*, ..., \mathbf{x}_{n^*}^*\}$ korral

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

3. M-samm: uuendame nii C teadaoleva kui ka $K - C$ uue klassi parameetrid, kasutades $\gamma(z_{nk})$ väärtuseid, mis on arvutatud E-sammul

$$\boldsymbol{\mu}_k^{uus} = \frac{1}{N_k + N_k^*} \left(\sum_{n=1}^{N} z_{nk}\mathbf{x}_n + \sum_{n=1}^{N^*} \gamma(z_{nk})\mathbf{x}_n \right)$$

$$\boldsymbol{\Sigma}_k^{uus} = \frac{1}{N_k + N_k^*}(\mathbf{S}_k + \mathbf{S}_k^*)$$

$$\pi_k^{uus} = \frac{N_k + N_k^*}{N + N^*}$$

kus

$$N_k = \sum_{n=1}^{N} z_{nk}$$

$$N_k^* = \sum_{n=1}^{N^*} \gamma(z_{nk})$$

$$\mathbf{S}_k = \sum_{n=1}^{N} z_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k^{uus})(\mathbf{x}_n - \boldsymbol{\mu}_k^{uus})^T$$

$$\mathbf{S}_k^* = \sum_{n=1}^{N^*} \gamma(z_{nk})(\mathbf{x}_n^* - \boldsymbol{\mu}_k^{uus})(\mathbf{x}_n^* - \boldsymbol{\mu}_k^{uus})^T$$

4. Hinda log-tõepära (valem 1.9) ja kontrolli log-tõepära koonduvust. Kui tõepära koonduvuse kriteerium pole täidetud, mine sammu 2 juurde tagasi

Pseudokood 4 Induktiivne mudel

1. Algväärtusta $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k ja hinda log-tõepära (valem 1.10).
2. E-samm: testandmete $\{\mathbf{x}_1^*, ..., \mathbf{x}_{n^*}^*\}$ põhjal arvutame

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

3. M-samm: uuendame $K-C$ uue klassi parameetrid, kasutades $\gamma(z_{nk})$, mis on arvutatud E-sammul. Teadaolevate klasside parameetrid jäävad samaks (välja arvatud komponentide osakaalud).

$$\boldsymbol{\mu}_k^{uus} = \frac{1}{N_k^*} \sum_{n=1}^{N^*} \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{uus} = \frac{1}{N_k^*} \sum_{n=1}^{N^*} \gamma(z_{nk})(\mathbf{x}_n^* - \boldsymbol{\mu}_k^{uus})(\mathbf{x}_n^* - \boldsymbol{\mu}_k^{uus})^T$$

teadaolevate klasside korral:

$$\pi_k^{uus} = (1 - \sum_{l=C+1}^{K} \frac{N_l^*}{N^*}) \frac{N_k}{N}$$

uute klasside korral:

$$\pi_k^{uus} = \frac{N_k^*}{N^*}$$

kus

$$N_k = \sum_{n=1}^{N} z_{nk}$$

$$N_k^* = \sum_{n=1}^{N^*} \gamma(z_{nk})$$

4. Hinda log-tõepära (valem 1.10) ja kontrolli log-tõepära koonduvust. Kui tõepära koonduvuse kriteerium pole täidetud, mine sammu 2 juurde tagasi.

1.3.2 Mudelivalik

Gaussi segumudeli treenimisel peame määrama klasside arvu K. See on teadmata ning treenides mudeleid erinevate K korral peame valima välja parima mudeli. Kuidas seda teha? Logaritmilise tõepära põhjal ei saa me otsuseid teha, sest saame tõepära parandada, kui lisame mudelile aina rohkem parameetreid. Selliselt käitudes võime mudeli ületreenida ning on oht, et mudel ei kirjelda tulevasi andmepunkte hästi. Seega peaksime hinnatavate parameetrite arvu kuidagi arvesse võtma.

Klassikalised võtted mudelivalikuks segumudelite kontekstis on penaliseeritud log-tõepära kriteeriumid nagu Akaike informatsioonikriteerium (AIC) ja Bayesi informatsioonikriteerium (BIC). BIC kriteerium toob sisse karistustingimuse parameetrite arvule. BIC üldkuju on järgmine:

$$BIC(\mathcal{M}) = \ln p(\mathbf{x}_1, ..., \mathbf{x}_n|\boldsymbol{\theta}) - \frac{\nu(\mathcal{M})}{2} \ln(n) \ , \tag{1.11}$$

kus $\nu(\mathcal{M})$ on hinnatavate parameetrite arv mudeli $\mathcal{M}$ korral, n on andmepunktide arv.

Transduktiivsel juhul on parameetrite arv mudelis võrdne $(K-1) + KD + KD(D+1)/2$, induktiivsel juhul aga $(K-1) + (K-C)D + (K-C)D(D+1)/2$. Märgime, et liidetavad tähistavad vastavalt osakaalude, keskmiste ja kovariatsioonimaatriksite hindamiseks vajaminevaid parameetrite arvu.

Pseudokood 5 Mudelivalik

1. Määra maksimaalne uute klasside arv S, mida mudel andmetest kontrollib.
2. $j = 0, 1, ..., S$ korral korda samme a ja b kuni log-tõepära pole 5 järjestikul korral paranenud:
    a. Initsialiseeri algparameetrid $\boldsymbol{\theta}$.
    b. Treeni mudel j uue klassiga.
4. Vali parima BIC väärtusega mudel.

Peatükk 2

Dirichlet' protsessi segumudel

Eelmises peatükis kirjeldatud meetodites kasutasime Bayesi informatsioonikriteeriumi klastrite arvu K valimiseks. Selles peatükis uurime alternatiivset viisi mudelivaliku tegemiseks. Nimelt pöördume mitteparameetrilise Bayesi statistika valdkonda: anname parameetritele eeljaotused ning lubame mudeli keerukusel kasvada, kui andmemaht kasvab. Selle saavutame, kui Bayesi segumudeli korral kasutame komponentide osakaalude eeljaotusena Dirichlet' protsessi. Peatükk on kirjutatud peamiselt järgmiste allikate põhjal Teh et al. [2005], Frigyik et al. [2010], Teh [2010], Sudderth [2006], Neal [2006], Görür [2007].

2.1 Dirichlet' protsess

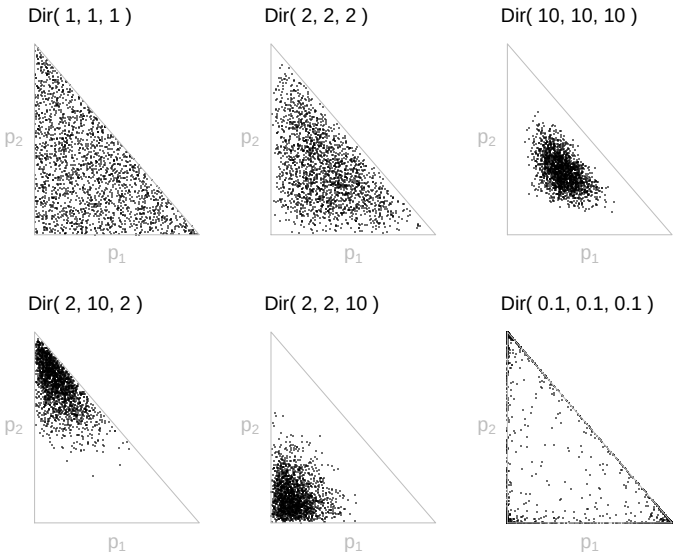
Ühekomponendiline normaaljaotus ei ole sobilik reaalsete andmete modelleerimiseks. Lõpliku arvu komponentidega segumudel kirjeldab andmeid paremini, kuid veelgi parem on mudel ülimalt loenduva arvu komponentidega, kus parameetrite eeljaotus on võimalikult paindlik ja mudeli keerukus sõltub vastavalt andmemahule. Dirichlet' protsess on jaotus, mis aitab meil sellist mudelit konstrueerida. Enne Dirichlet' protsessi defineerimist anname vaistliku ülevaate Dirichlet' ja beeta jaotusest. Need on vajalikud Dirichlet' protsessi mõistmiseks, sest Dirichlet' protsess on lõpmatumõõtmeline üldistus Dirichlet' jaotusest, mis omakorda on mitmemõõtmeline üldistus beeta jaotusest.

2.1.1 Seotud jaotused - beeta ja Dirichlet'

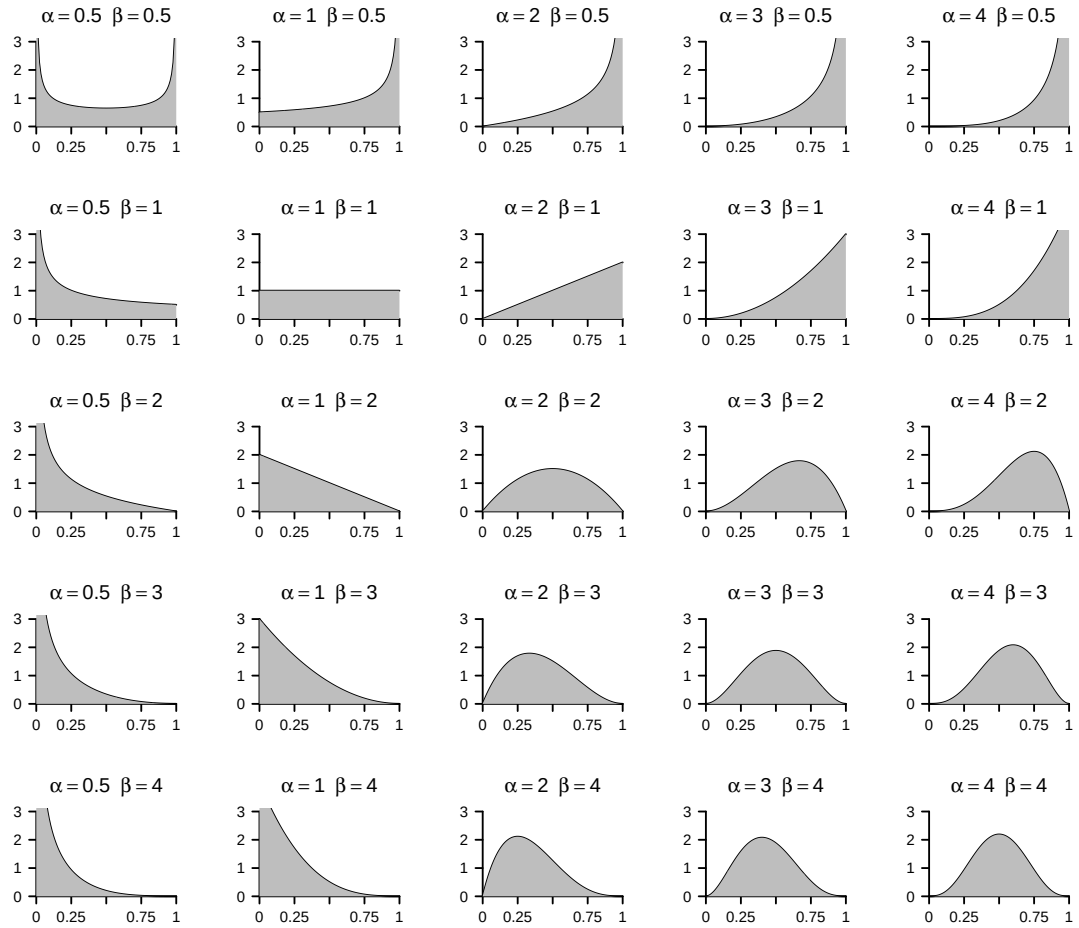
Öeldakse, et juhuslik suurus X on Dirichlet' jaotusega parameetriga $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)$ ja tähistatakse $X \sim \text{Dir}(\boldsymbol{\alpha})$, kui tema tihedusfunktsioon avaldub kujul

$$f(\mathbf{x}|\boldsymbol{\alpha}) = C \prod_{i=1}^{K} x_i^{\alpha_i - 1} , \tag{2.1}$$

kus C on normeeriv konstant, $x_i \in [0, 1]$ ja $\sum_{i=1}^{K} x_i = 1$. Kui $K = 2$, siis Dirichlet' jaotus taandub beeta jaotuseks. Märkame, et Dirichlet' jaotuse funktsionaalne vorm on sarnane multinomiaaljaotusele. Bayesi paradigmas kasutatakse Dirichlet' jaotust sageli multinomiaaljaotuse eeljaotusena, sest ta on kaasjaotus (*conjugate prior*) sellele. Joonisel 2.1 on toodud näited Dirichlet' jaotusest $K = 3$ korral, joonisel 2.2 on näited beeta jaotusest erinevate $\boldsymbol{\alpha}$ väärtuste korral. Kuna Dirichlet' jaotus on defineeritud $K - 1$ simpleksil, siis võime öelda, et Dirichlet jaotus on jaotus üle jaotuste.



Joonis 2.1: Näited erineva parameetri $\boldsymbol{\alpha}$ väärtustega Dirichlet' jaotustest, kui $K = 3$, mis on visualiseeritud 2D simpleksil. Suuremad $\boldsymbol{\alpha}$ väärtused kontsentreerivad tõenäosusjaotust, võrdsete $\alpha_1, \alpha_2, \alpha_3$ korral on jaotus sümmeetriline.



Joonis 2.2: Näited erinevatest beeta jaotustest. Kasutame tähistust $\boldsymbol{\alpha} = (\alpha, \beta)$. Kui $\alpha = 1$ ja $\beta = 1$, siis on tegemist ühtlase jaotusega lõigus [0, 1]. Suuremad α ja β väärtused (seega väiksem dispersioon) kontsentreerivad tõenäosusjaotust valitud keskmisele. Kui α ja β on nullilähedased, siis tõenäosusjaotus on kontsentreerunud lõigu [0, 1] äärealadele. Märgime, et beeta jaotuse keskväärtus ja dispersioon avalduvad järgmiselt: $E(X) = \frac{\alpha}{\alpha+\beta}$ ja $D(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

2.1.2 Dirichlet' protsessi definitsioon

Järgnevalt anname formaalse kirjelduse Dirichlet' protsessile. Selleks on vaja sisse tuua mõned mõisted mõõduteooriast. Olgu X hulk ja $\mathcal{B}$ olgu σ-algebra hulgal X. See tähendab, et $\mathcal{B}$ on X-i alamhulkade kogum, nii et on täidetud järgmised nõuded:

(1) $X \in \mathcal{B}$

(2) $A \in \mathcal{B} \Rightarrow A^c \in \mathcal{B}$

(3) $A_j \in \mathcal{B}, j \in \mathbb{N} \Rightarrow \cup_{j=1}^{\infty} A_j \in \mathcal{B}$

Me nimetame hulgafunktsiooni $\mu : \mathcal{B} \to [0, \infty]$ mõõduks, kui $\mu(\emptyset) = 0$ ja μ on σ-aditiivne. Kusjuures me ütleme, et hulgafunktsioon μ on σ-aditiivne (ehk loenduvalt aditiivne), kui paarikaupa lõikumatute hulkade $A_j \in \mathcal{B}, j \in \mathbb{N}$ korral kehtib $\mu(\cup_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} \mu(A_j)$. Kolmikut $(X, \mathcal{B}, \mu)$ kutsume mõõduga ruumiks. Kui $\mu(X) = 1$, siis mõõtu μ nimetatakse tõenäosusmõõduks ja $(X, \mathcal{B}, \mu)$ tõenäosusruumiks. Sel juhul kutsutakse X-i elementaarsündmuste ruumiks (juhusliku katse kõik võimalikud tulemused) ja $\mathcal{B}$ elemente nimetatakse sündmusteks. Näiteks täringuviske korral on $X = \{1, 2, 3, 4, 5, 6\}$ ja

$$\mathcal{B} = \{\emptyset, \{1\}, \ldots, \{6\}, \{1, 2\}, \ldots, \{5, 6\}, \ldots, X\} \, . \tag{2.2}$$

Me ütleme, et juhuslik tõenäosusmõõt G on Dirichlet' protsessi jaotusega kontsentratsiooni parameetritega α ja baasjaotusega G_0 ning tähistame $G \sim DP(\alpha, G_0)$, kui paarikaupa lõikumatute $A_1, \ldots, A_K$, $A_1 \cup \ldots \cup A_K = X$ korral $(G(A_1), ..., G(A_K)) \sim \text{Dir}(\alpha G_0(A_1), ..., \alpha G_0(A_K))$.

Dirichlet' protsessil on kaks parameetrit: baasjaotus G_0 ja kontsentratsiooni parameeter α. Baasjaotust võime intuitiivselt interpreteerida kui keskmist, kontsentratsiooni parameetrit kui täpsust. Tõepoolest, iga mõõtuva regiooni $T \subset X$ korral $E(G(T)) = G_0(T)$, kus $G \sim DP(\alpha, G_0)$. Seda saab näidata kasutades Dirichlet' jaotuse keskväärtuse valemit ja Dirichlet' jaotuse grupeerimisomadust. Analoogiliselt, kasutades Dirichlet' jaotuse dispersiooni valemit, saame et

$$D(G(T)) = \frac{G_0(T)(1 - G_0(T))}{\alpha + 1} \, . \tag{2.3}$$

Seega mida suurem on α, seda väiksem on dispersioon ja Dirichlet' protsess sätib

suurema osa tõenäosusmassist keskmise ümber. Paneme tähele, et α ja G_0 esinevad Dirichlet' protsessi definitsioonis ainult korrutisena, seega defindeerides $G_0^* := \alpha G_0$ on võimalik kasutada αG_0 kui ainult üht parameetrit ja kirjutada $DP(G_0, \alpha)$ asemel $DP(G_0^*)$. Sellise parametriseerimisega kaotaks aga α ja G_0 oma tähenduse DP kirjeldamisel (Frigyik et al. [2010]).

2.1.3 Jaotusest genereerimine

Eelnevalt kirjeldasime küll formaalselt, mida tähendab Dirichlet' protsess, kuid me ei oska juhuslikke suurusi sellest jaotusest genereerida. Ja kas üldsegi leidub selline mõõt, mis rahuldab eelnevat definitsiooni? Dirichlet' protsessil on mitmeid generatiivseid kirjeldusi. Järgnevalt on toodud näited, kuidas genereerida realisatsioone Dirichlet' protsessist. Täpsemalt vaatleme toki murdmise protsessi, Hiina restorani protsessi ja Polya urni skeemi, millest kaks viimast on erinevad nimetused samale protsessile. Need konstruktsioonid omavad tähtsat rolli Dirichlet' protsessi arvutuslike meetodite kirjeldamisel. Kõigi nende skeemide korral genereerime jadad $\{\pi_k\}$ ja $\{\boldsymbol{\theta}_k\}$ ning tähistame

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \ , \tag{2.4}$$

et saada juhuslik diskreetne mõõt G. Parameetrid $\{\boldsymbol{\theta}_k\}$ tõmmatakse tavaliselt baasjaotusest G_0. Veidi raskem on saada juhuslikke $\{\pi_k\}$. Märgime, et δ_x-ga tähistame Diraci mõõtu, mis on võrdne 1-ga, kui x kuulub huvipakkuvasse hulka ja 0 vastasel korral.

Järgnevalt kirjeldame toki murdmise skeemi:

$$\begin{aligned} \beta_k &\sim \text{Beta}(1, \alpha) \ , \\ \boldsymbol{\theta}_k &\sim G_0 \ , \\ \pi_k &= \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \ , \\ G &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \ . \end{aligned} \tag{2.5}$$

Sethuraman [1994] näitas, et selliselt toimides on juhuslik tõenäosusmõõt G tõesti jaotusest $DP(\alpha, G_0)$. Intuitiivselt saame seda protsessi tõlgendada järgmiselt:

- Olgu meil tokk pikkusega 1.
- Genereerime juhusliku suuruse $\beta_1 \sim \text{Beta}(1, \alpha)$.
- Murrame toki katki kohalt β_1, anname π_1 väärtuseks vasakule jääva toki pikkuse (ehk β_1).
- Nüüd võtame paremale jäänud toki, genereerime $\beta_2 \sim \text{Beta}(1, \alpha)$, murrame toki katki kohalt β_2 ja seega saame π_2 väärtuseks $(1 - \beta_1)\beta_2$.
- Jätkame selliselt.

Selline Dirichlet' protsessi kirjeldus annab meile võimaluse tõlgendada parameetrit α. Kuna proportsioonid $\beta_k \sim \text{Beta}(1, \alpha)$, siis teame, et

$$E(\beta_k) = \frac{1}{1+\alpha} \, . \tag{2.6}$$

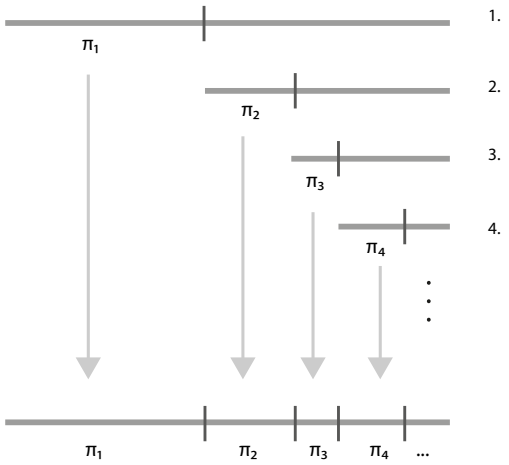
Seega väikeste α väärtuste korral on suurem osa tõenäosusmassist jaotunud paarile esimesele komponendile. Joonisel 2.3 on näidatud tokimurdmise skeem, joonisel 2.4 erineva α väärtuste korral saadud toki pikkused ja joonisel 2.5 on erinevate baasjaotuste korral tokkidele vastavusse pandud parameetrid $\boldsymbol{\theta}_k$.

Teine perspektiiv on Dirichlet' protsessi vaadata läbi Polya urni skeemi. Oletame, et genereerime lõptmatu jada $\{\boldsymbol{\theta}_i\}$ järgmise eeskirja alusel:

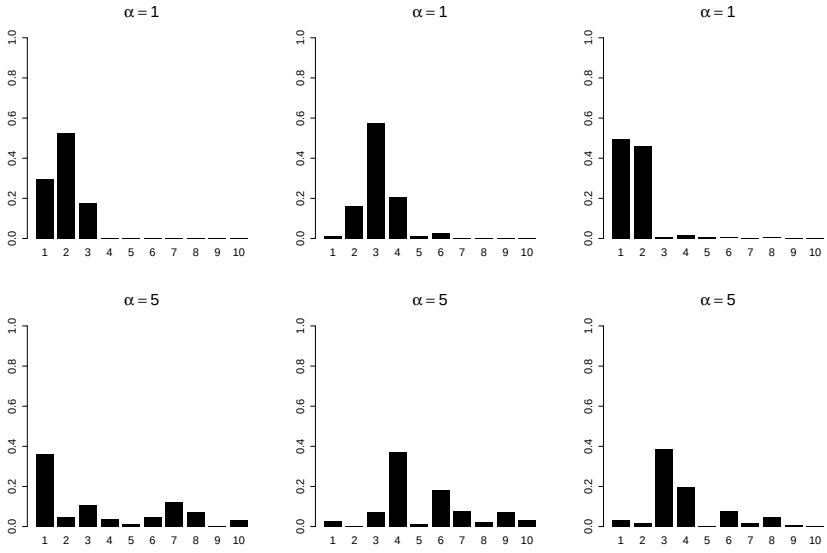
$$\begin{aligned} \boldsymbol{\theta}_1 &\sim G_0 \, , \\ \boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n &\sim G_n(\boldsymbol{\theta}_{n+1}) = \frac{\alpha G_0 + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n} \, . \end{aligned} \tag{2.7}$$

Metafooriliselt võime mõelda sellest kui protsessist, kus tõmbame värvilisi palle urnist G, kus $\boldsymbol{\theta}_i$ tähistab i-nda palli värvi, mille urnist võtsime.

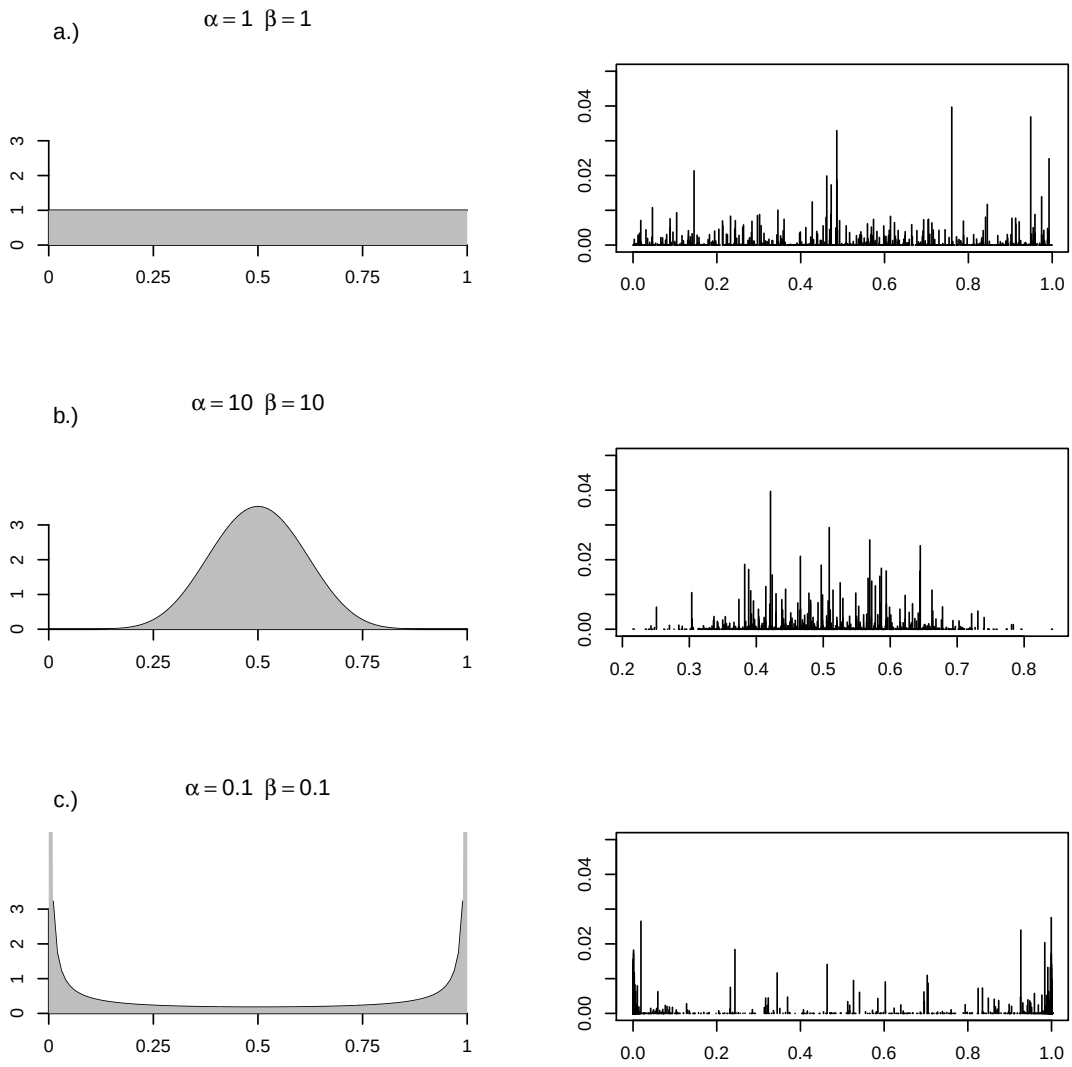
- Meil on urn, milles on algselt kokku α palli, kus $\mathbf{x}$ värviga palle on $\alpha G_0(\mathbf{x})$ (märgime, et pallide arv võib olla ka murdarv)
- Igal sammul võtame urnist palli, teeme kindlaks selle värvi ning paneme selle palli koos veel ühe sama värvi palliga urni tagasi.



Joonis 2.3: Iteratiivne protseduur π_k-de saamiseks kasutades toki murdmise skeemi. Vertikaalsed jooned näitavad toki murdmise kohti ja on saadud simuleerides jaotusest Beta$(1, \alpha)$. Toki pikkustest saame π_k väärtused.



Joonis 2.4: Toki murdmise protsess $\alpha = 1$ ja $\alpha = 5$ korral. Väiksema α korral on tõenäosusjaotus kontsentreerunud vähematele komponentidele.



Joonis 2.5: Jaotused, mis on genereeritud Dirichlet' protsessist $\alpha = 100$ ja erinevate baasjaotuse korral läbi toki murdmise skeemi. Baasjaotustena on kasutatud erinevate parameetritega beeta jaotuseid.

Kuidas selline eeskiri seostub Dirichlet' protsessiga? Blackwell and MacQueen [1973] on näidanud, et kui jätkame seda protsessi lõpmatuseni, siis G_n koondub peaaegu kindlasti juhuslikuks diskreetseks jaotuseks G, mis on jaotusest $DP(\alpha, G_0)$. Selle saame formaalselt kirjutada:

$$\lim_{n\to\infty} G_n \to G \sim DP(\alpha, G_0) \ . \tag{2.8}$$

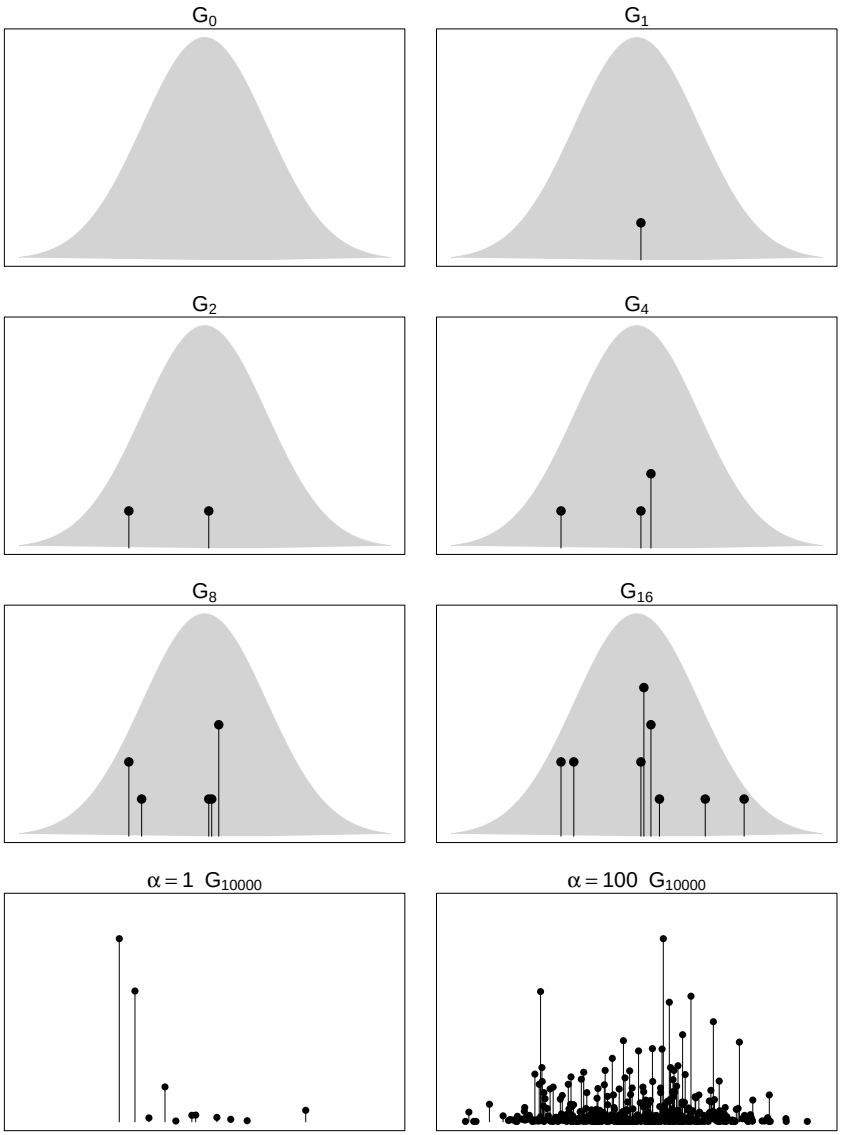
Lisaks sellele on $\{\boldsymbol{\theta}_i\}, i = 1, ..., n$, valim jaotusest G.

Polya urni skeem ilmestab Dirichlet' protsessi klasterdamisomadust: positiivse tõenäosusega on urnist tõmmatud kaks palli sama värviga, seega võime andmepunktid jaotada värvide põhjal gruppidesse. Indekseerime erinevad värvid täisarvudega ja tähistame c_i-ga i-nda palli grupiindeksit. Seega kui i-s ja j-s pall on sama värvi, siis $c_i = c_j$. Märgime, et c_i erineb $\boldsymbol{\theta}_i$-st, sest kui $\boldsymbol{\theta}_i$ on palli värv, siis c_i tähistab värvi gruppi. Oletame nüüd, et oleme urnist võtnud N palli ja näinud K erinevat värvi. Valemist 2.7 järeldub nüüd, et

$$p(c_{n+1}|c_1, ..., c_n) = \frac{\alpha}{\alpha + n}\delta_{K+1} + \sum_{k=1}^{K} \frac{n_k}{\alpha + n}\delta_k \ , \tag{2.9}$$

kus n_k tähistab pallide arvu värvidegrupis k. Seega järgmise palli värv on tõenäosusega $\frac{n_k}{\alpha+n_k}$ grupi k värv või tõenäosusega $\frac{\alpha}{\alpha+n_k}$ mingi uus värv. Kasutades teistsugust metafoori, saame Polya urni skeemiga ekvivalentse protsessi nimega Hiina restorani protsess.

- Algselt on restoran tühi.
- Esimene külastaja istub esimese laua taha.
- Teine külastaja istub tõenäosusega $\frac{\alpha}{\alpha+1}$ uue laua taha, tõenäosusega $\frac{1}{\alpha+1}$ istub ta esimese külastaja kõrvale.
- ...
- $n + 1$ külastaja istub uue laua taha tõenäosusega $\frac{\alpha}{\alpha+n}$, k-nda laua taha tõenäosusega $\frac{n_k}{\alpha+n_k}$, kus n_k on inimeste arv, kes praegu k-nda laua taga söövad.

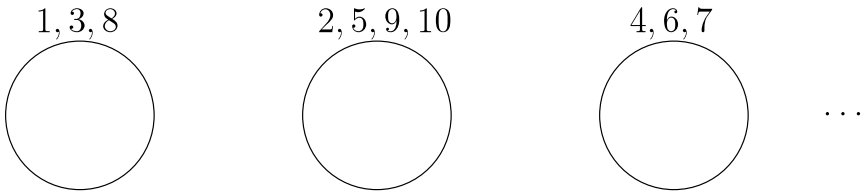


Joonis 2.6: Järjestikused genereerimised Polya urni protsessist. Baasjaotuseks on standard normaaljaotus. Esimesed kolm rida näitavad G_n evolutsiooni $\alpha = 5$ korral. Viimane rida näitab G_n pärast 10000 palli võtmist urnist $\alpha = 1$ ja $\alpha = 100$ korral.

Joonisel 2.7 on toodud üks võimalus, kuidas 10 esimest klienti võivad restoranis laudade vahel jaguneda. Sellise asetuse tõenäosus on

$$p(z_1, ..., z_{10}) = p(z_1)p(z_2|z_1)...p(z_{10}|z_1, ..., z_9) = \\ = \frac{\alpha}{\alpha}\frac{\alpha}{1+\alpha}\frac{1}{2+\alpha}\frac{\alpha}{3+\alpha}\frac{1}{4+\alpha}\frac{1}{5+\alpha}\frac{2}{6+\alpha}\frac{2}{7+\alpha}\frac{2}{8+\alpha}\frac{3}{9+\alpha} . \tag{2.10}$$

Paneme tähele, et sellise jagunemise tõenäosus ei sõltu inimeste tulemise järjekorrast. Vahetades klientide tulemise järjekorda muutub küll eelneva avaldise lugeja, kuid nimetaja jääb samaks. Sellist omadust kutsutakse vahetatavuseks (*exchangeability*) ja see on kasulik Dirichlet' protsessi segumudeli parameetrite hindamise algoritmides.



Joonis 2.7: Hiina restorani protsess. Üks võimalik kümne kliendi jagunemine laudade vahel. Ringid tähistavad laudu ja numbrid tähistavad kliente, kes istuvad vastavas lauas. Joonis võetud: Blei [2007]

2.1.4 Segumudeli kirjeldus

Dirichlet' protsessi segumudel on segumudel ülimalt loenduva arvu komponentidega. Me modelleerime andmeid $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ kasutades parameetreid $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n\}$. Iga $\boldsymbol{\theta}_i$ on genereeritud jaotusest G, iga $\mathbf{x}_i$ on jaotusest $F(\boldsymbol{\theta}_i)$ parameetriga $\boldsymbol{\theta}_i$. Kirjeldame generatiivse mudeli:

$$\begin{aligned} G|\alpha, G_0 &\sim DP(\alpha, G_0) , \\ \boldsymbol{\theta}_i|G &\sim G , \\ \mathbf{x}_i|\boldsymbol{\theta}_i &\sim F(\boldsymbol{\theta}_i) . \end{aligned} \tag{2.11}$$

Kuna G on diskreetne, siis mitmed $\boldsymbol{\theta}_i$ võivad võtta sama väärtuse ning eelnevat mudelit saab tõlgendada kui segumudelit, kus $\mathbf{x}_i$-d, millel on sama $\boldsymbol{\theta}_i$, kuuluvad samasse klastrisse. Segumudeli perspektiivi saab kergemini esile tuua kasutades toki murdmise konstruktsiooni. Olgu z_i klastri kuuluvuse näitaja, mille väärtuseks on k tõenäosusega π_k. Seega eelnevalt kirjeldatud mudeli saame ekvivalentselt kirjutada kujul:

$$\begin{aligned} \boldsymbol{\pi}|\alpha &\sim \text{tokimurdmine}(\alpha) \ , \\ c_i|\boldsymbol{\pi} &\sim \text{Mult}(\boldsymbol{\pi}) \ , \\ \boldsymbol{\theta}_k^*|G_0 &\sim G_0 \ , \\ \mathbf{x}_i|c_i, \{\boldsymbol{\theta}_k^*\} &\sim F(\boldsymbol{\theta}_{z_i}^*) \end{aligned} \tag{2.12}$$

ja tähistame $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_{c_i}^*}$ ja $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{c_i}^*$. Sarnaselt segumudeli kirjeldusega eelmises peatükis: $\boldsymbol{\pi}$ on klastrite osakaal, $\boldsymbol{\theta}_k^*$ on klastri parameetrid ja $F(\boldsymbol{\theta}_k^*)$ on jaotus üle k-nda klastri andmete. G_0 on aga eeljaotus üle parameetrite. DP segumudel on lõpmatu segumudel - segumudel, millel on ülimalt loenduv arv klastreid. Kuna $\boldsymbol{\pi}_k$ suurused vähenevad kiiresti, siis andmete modelleerimiseks kasutatakse ainult väikest osa klastritest. See on suur erinevus võrreldes lõplike segumudelitega - klastrite arv ei ole fikseeritud ja seda on võimalik järeldada andmetest. Seega saame kõrvale hiilida mudelivalikust.

2.2 Parameetrite hindamine

Olgu meil andmed $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$. Soovime teada saada klastrite arvu K ja nende klastrite parameetreid $\boldsymbol{\theta}_k$. Analüütiline parameetrite hindamine ei ole Dirichlet' segumudeli puhul võimalik, seega tuleb kasutada lähendusmeetodeid nagu variatsiooniline Bayes või MCMC meetodid. Selles töös kasutati parameetrite hindamiseks MCMC meetodit nimega Gibbsi valik.

2.2.1 Gibbsi valik

Olgu $p(X_1, ..., X_K)$ huvipakkuv jaotus, millest soovime andmepunkte simuleerida. Oletame, et sellest ühisjaotusest on simuleerimine keeruline või võimatu, seevastu

tinglikest jaotustest $p(X_j|X_1, ..., X_{j-1}, X_{j+1}, ..., X_K)$ on simuleerimine kerge $j = 1, 2, ..., K$ korral. Gibbsi valik on MCMC meetod, mis simuleerib järgemööda neist tinglikest jaotustest, et saada soovitud ühisjaotusest $p(X_1, ..., X_K)$ andmepunkte. Gibbsi valiku algoritm on näidatud pseudokoodis 6.

Pseudokood 6 Üldine Gibbsi valiku algoritm

1. Anname algväärtused $X_1^0, X_1^0, ..., X_K^0$
2. Genereerime uued väärtused järgmiselt:

$$
\begin{aligned}
X_1^{t+1} &\sim p(X_1|X_2^t, X_3^t, ..., X_K^t) \\
X_2^{t+1} &\sim p(X_2|X_1^{t+1}, X_3^t, ..., X_K^t) \\
&\dots \\
X_K^{t+1} &\sim p(X_K|X_1^{t+1}, X_2^{t+1}, ..., X_{K-1}^t)
\end{aligned}
$$

Paneme tähele, et Gibbsi valik on EM-algoritmiga lähedalt seotud: suurimaks erinevuseks on asjaolu, et Gibbsi valik tõmbab realisatsioone tinglikest jaotustest, EM-algoritm maksimiseerib neid.

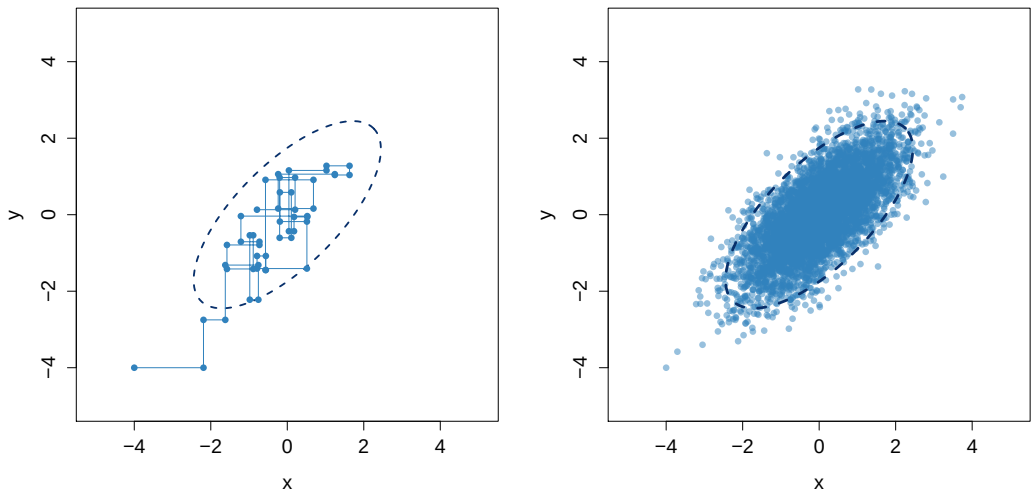
Kirjeldame Gibbsi valikut lihtsa näite põhjal. Oletame, et soovime simuleerida andmepunkte kahemõõtmelisest normaaljaotusest $(X_1, X_2) \sim \mathcal{N}(\mu, \Sigma)$, kus $\mu = (0, 0)$ ja $\Sigma = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$. Oletame, et me ei oska andmepunkte genereerida sellest kahemõõtmelisest jaotusest, seevastu teame, et tinglikud jaotused on ühemõõtmelised normaaljaotused ja neist oskame andmepunkte simuleerida. Nimelt saame X_2 simuleerida järgmiselt:

$$X_2 \sim \mathcal{N}(\rho X_1, 1 - \rho^2) \ . \tag{2.13}$$

Analoogiliselt genereeritakse ka X_1 väärtused. Joonisel 2.8 on näidatud 25 esimest Gibbsi valiku iteratsiooni, lisaks sellele on illustreeritud Gibbsi valikuga genereeritud 5000 andmepunkti sellest jaotusest.

2.2.2 Gibbsi valik Dirichlet' protsessi segumudeli jaoks

Eelnevalt kirjeldasime erinevaid analooge, kuidas Dirichlet' protsessist vaatluseid genereerida. MCMC meetodid Dirichlet' protsessi jaoks jagunevad laias laastus



Joonis 2.8: Kahemõõtmelisest normaaljaotusest simuleerimine Gibbsi valiku abil. Vasakul on näidatud Gibbsi valiku 25 esimest iteratsiooni, paremal on näidatud Gibbsi valikuga genereeritud 5000 andmepunkti.

kaheks: ühed kasutavad Polya urni esitust ja simuleerivad parameetreid $\boldsymbol{\theta}_i$, teised kasutavad jaotust G läbi toki murdmise skeemi. Me vaatame lähemalt algoritme, mis kasutavad Polya urni esitust ja kasutavad eeljaotustena kaasjaotusi.

Polya urni skeemi korral genereerime $\boldsymbol{\theta}_i$ väärtuseid jaotusest G, kuigi me ei tea G täpset esitust. Kasutades Polya urni skeemi, saame Dirichlet' segumudeli (2.11) esitada kujul:

$$\begin{aligned} \boldsymbol{\theta}_i|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{i-1} &\sim G_n \ , \\ \mathbf{x}_i|\boldsymbol{\theta}_i &\sim F(\boldsymbol{\theta}_i) \ . \end{aligned} \tag{2.14}$$

kus G_n on defineeritud valemis 2.7.

Kõige lihtsama parameetrite hindamise skeemi korral defineeritakse Markovi ahelaks parameetrid $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$. Iga parameetrit uuendatakse tinglikust järeljaotusest, mis on antud läbi Polya urni skeemi. Kasutades andmepunktide vahetatavuse omadust, võime alati oletada, et $\boldsymbol{\theta}_i$ on viimane andmepunkt ja võime tinglikustada üle kõigi teiste parameetrite väärtuste.

Oletades, et $\boldsymbol{\theta}_i$ on viimane vaatlus N andmepunktist, saame $\boldsymbol{\theta}_i$ eeljaotu-

se, tinglikustades üle kõigi teiste parameetrite, kirjutada läbi Polya urni skeemi järmiselt:

$$p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}) = \frac{\alpha}{\alpha+N-1}G_0 + \frac{1}{\alpha+N-1}\sum_{i\neq j}\delta_{\theta_j} \ , \tag{2.15}$$

kus $\boldsymbol{\theta}_{-i}$ tähistab kõiki parameetreid välja arvatud $\boldsymbol{\theta}_i$. Kombineerides seda tõepäraga $F(\mathbf{x}_i|\boldsymbol{\theta})$, saame tingliku järeljaotuse kirjutada järgmiselt:

$$p(\boldsymbol{\theta}_i|\mathbf{x}_i, \boldsymbol{\theta}_{-i}) \propto F(\mathbf{x}_i|\boldsymbol{\theta}_i)(\frac{\alpha}{\alpha+N-1}G_0 + \frac{1}{\alpha+N-1}\sum_{i\neq j}\delta_{\theta_j}) \ . \tag{2.16}$$

Selliselt toimides saadud Gibbsi valiku algoritm on toodud pseudokoodis 7.

Pseudokood 7 Gibbsi valik DP mudelile Polya urni skeemi kohaselt

Markovi ahela seisund koosneb parameetritest $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$.
1. $i = 1, ..., N$ korral uuenda $\boldsymbol{\theta}_i$ väärtust vastavalt valemile 2.16
2. korda sammu 1

Selle algoritmi koondumine on aeglane, sest $\boldsymbol{\theta}_i$ väärtused genereeritakse ühekaupa, kuigi teame, et paljud andmepunktid võivad sama parameetri väärtust omada. Seda algoritmi saab kiiremaks teha, kui kasutame Hiina restorani protsessi esitust. Me saame kasutada indikaatortunnuseid c_i näitamaks klastrisse kuuluvust ja $\boldsymbol{\theta}^*_k$-ga saame tähistada selle grupi parameetreid. Selle asemel, et uuendada iga $\boldsymbol{\theta}_i$ eraldi, võime uuendada c_i väärtuseid ja gruppide parameetreid $\boldsymbol{\theta}^*_k$.

Nüüd koosneb Markovi ahela olek indikaatortunnustest $c_1, ..., c_N$ ja komponentide parameetritest $\boldsymbol{\theta}^*_1, ..., \boldsymbol{\theta}^*_K$. Kasutades indikaatortunnuste eeljaotusena valemit 2.9 ning kombineerides seda tõepäraga, saame tinglikuks järeljaotuseks:

$$\begin{aligned} p(c_i = k|\mathbf{x}_i, \mathbf{c}_{-i}, \alpha, \boldsymbol{\theta}) &\propto \frac{n_{-i,k}}{N-1+\alpha}F(\mathbf{x}_i|\boldsymbol{\theta}_k) \ , \\ p(c_i = K+1|\mathbf{x}_i, \mathbf{c}_{-i}, \alpha) &\propto \frac{\alpha}{N-1+\alpha}\int F(\mathbf{x}_i|\boldsymbol{\theta})dG_0(\boldsymbol{\theta}) \ . \end{aligned} \tag{2.17}$$

Kui indikaatortunnused on uuendatud, tõmbame k-nda komponendi uued para-

meetrid järgmisest järeljaotusest:

$$p(\boldsymbol{\theta}_k^*|\mathbf{X}, \mathbf{c}) \propto \prod_{x_i \in klaster_k} F(\mathbf{x}_i|\boldsymbol{\theta}) G_0(\boldsymbol{\theta}) \ . \tag{2.18}$$

See Gibbsi valiku protseduur parameetrite järeljaotuse saamiseks on kokku võetud pseudokoodis 8.

Pseudokood 8 Gibbsi valik DP mudelile Hiina restorani skeemi kohaselt

1. Markovi ahela seisund koosneb indikaatortunnustest $\{c_1, ..., c_N\}$ ja komponentide parameetritest $\{\boldsymbol{\theta}_1^*, ..., \boldsymbol{\theta}_K^*\}$.
2. Uuenda c_i kasutades eelnevale toodud valemit 2.17 , $i = 1, ..., N$ korral. Kui $c_i = K + 1$, siis genereeri sellele komponendile uued parameetrid baasjaotusest G_0.
3. Uuenda $\boldsymbol{\theta}_i^*$ kasutades valemit 2.18, $i = 1, ..., K^*$ korral.
4. korda samme 2 ja 3

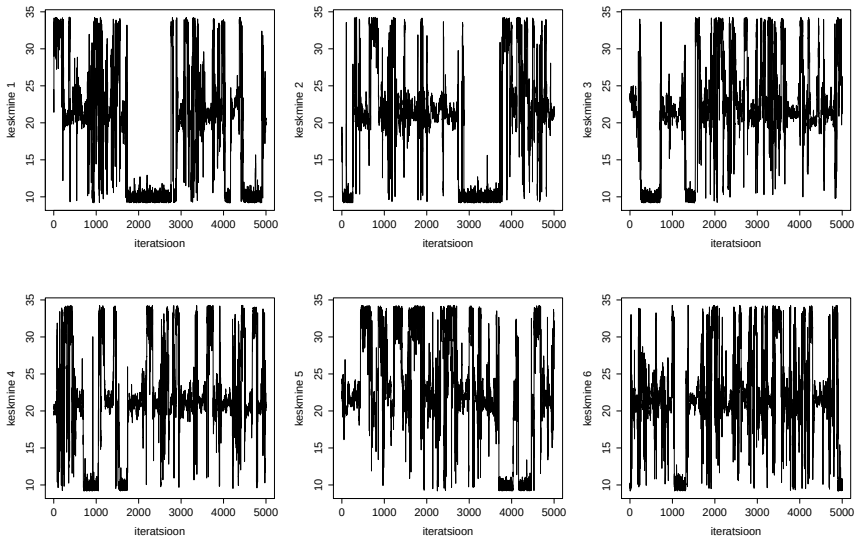
Näeme, et kui tahta rakendada seda algoritmi, on vaja, et eelnev integraal oleks analüütiliselt lahendatavad. Seega peab baasjaotus G_0 olema kaasjaotus tõepärale $F(\mathbf{X}|\boldsymbol{\theta})$. Kui tahta Dirichlet' protsessi segumudeliga modelleerida mitmemõõtmelisi andmestikke, siis on loomulik valik võtta tõepärafunktsiooniks on mitmemõõtmeline normaaljaotus. Mitmemõõtmelise normaaljaotuse kaasjaotuseks on Gaussi-Wisharti jaotus. Seega et vajamineva integraali väärtust leida, tuleb integreerida üle hirmuäratava Gaussi-Wisharti jaotuse Görür and Rasmussen [2010].

2.2.3 Siltide vahetumise probleem

Bayesi segumudelite korral on parameetrite hindamine raskendatud, sest parameetrite järeljaotus on tavaliselt sümmeetriline ja mitmetipuline ning tavaline praktika, kus parameetri hinnanguks määratakse järeljaotuse keskmine, ei tööta. Seda kutsutakse siltide vahetumise probleemiks. k-komponendilise segumudeli korral on $k!$ erinevat võimalust siltide määramiseks.

Illustreerime seda galaktika andmetel. Andmestik koosneb 82 galaktika kiirusest. Sobitasime andmestikule 6-komponendilise Bayesi segumudeli, kus eeljaotusena kasutasime nõrgalt informatiivset eeljaotust, mida on kasutatud ka

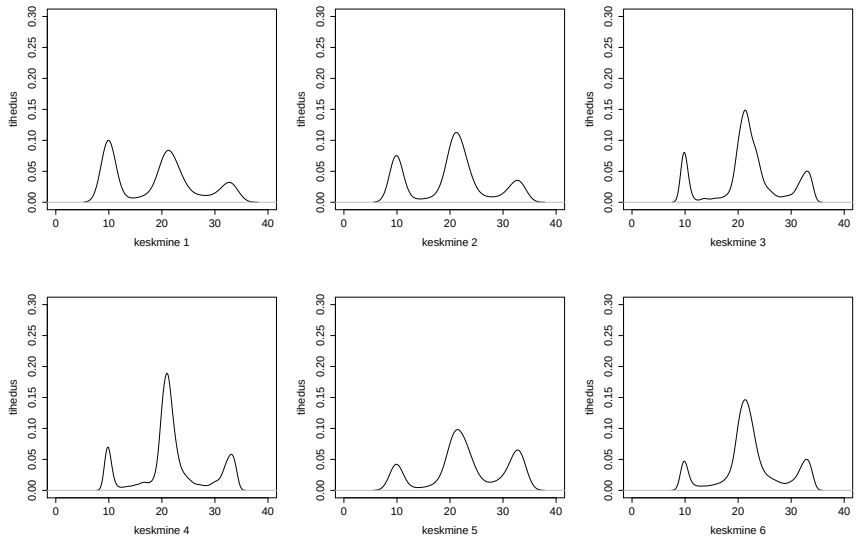
näiteks Raftery [1995]. Jooniselt 2.9 näeme, kuidas Gibbsi valiku jooksul klastrite keskmised vahetuvad ning vaadates klastrite keskmiste järeljaotuseid jooniselt 2.10, näeme, et klastrite keskmisi ei ole võimalik mõistlikult määrata, sest kõik järeljaotused on sarnased.



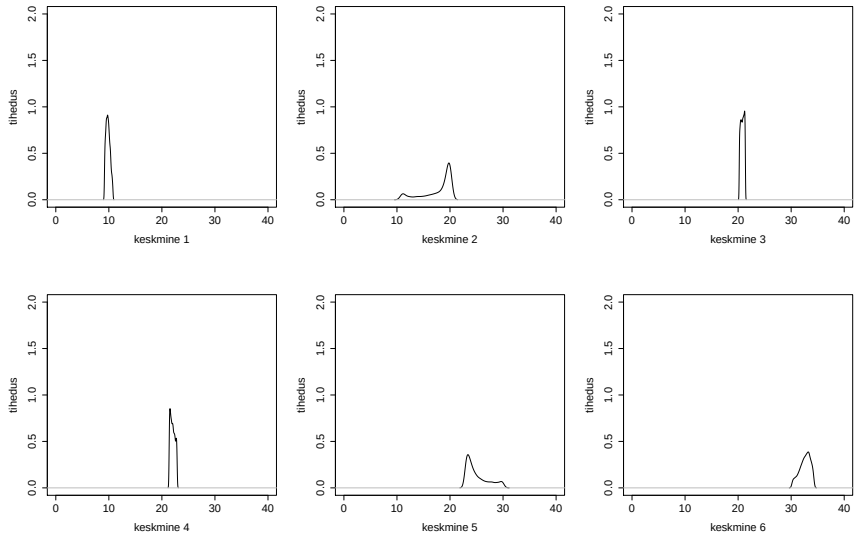
Joonis 2.9: Siltide vahetumise probleem. Näidatud on Gibbs valikul saadud klastrite keskmised erinevatel iteratsioonidel. Jooniselt on näha, kuidas klastrite keskmised hüppavad gruppide vahel Gibbsi valiku käigus.

Üks võimalik lahendus on kuntslikult tekitada identifitseeritus. Näiteks võime segu komponendid pärast iga Gibbsi tsüklit järjestada keskmiste alusel $\mu_1 \leq \mu_2 \leq ... \leq \mu_k$. Sel juhul saadud järeljaotused on toodud joonisel 2.11 ja näeme, et klastrite keskmiste hinnangud tulevad mõistlikumad kui enne. Kui tegemist on mitmemõõtmelise jaotusega, siis on keeruline aga keskmisi järjestada.

Võimalik on anda komponentidele ka tugev eeljaotus, mis muudab komponendid identifitseerituks. Meie implementeeritud segumudeli korral on teadaolevate klastrite korral see nõue küll tagatud, kuid uute klastrite korral mitte. Samuti on võimalik võrrelda andmepunktide kaugusi üksteisest ehk vaadata, kui tihti nad on samas klassis Gibbsi valiku erinevatel iteratsioonidel. Siis kandub probleem aga kaugusemaatriksi tõlgendamisele. Samuti võib Dirichlet' protsessi segumudeli põhjal otsustada vaid erinevate klastrite arvu ning klastrite parameetrite



Joonis 2.10: Siltide vahetumise probleem. Klastrite keskmiste järeljaotused on sarnased, parameetreid pole võimalik mõistlikult hinnata.



Joonis 2.11: Siltide vahetumise probleem. Klastri keskmiste järeljaotus, kui tekitasime kuntsliku identifitseerituse $\mu_1 \leq \mu_2 \leq ... \leq \mu_k$. Nüüd on võimalik klastri keskmisi mõistlikumalt hinnata.

saamiseks sobitada andmetele klassikaline segumudel. Dirichlet' protsessi segumudeli testimisel lahendasime kergematel juhtudel siltide vahetumise probleemi järjestades komponentide keskmised, muudel juhtudel järeldasime vaid komponentide arvu.

2.2.4 Pseudokood

Pseudokood 9 Dirichlet protsessi segumudeli parameetrite hindamise algoritm

Oleme Markovi ahela olekus t, kus klassikuuluvuse indeksid on $c_1, ..., c_n$ ja klastrite parameetrid on $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$. Tahame genereerida järgmise oleku (oleku t+1) vajalikud väärused ($c_1, ..., c_n$ ja $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$)

Uuendame indikaatortunnused (i=1,...,n korral)
1. eemaldame andmepunkti $\mathbf{x}_i$ praegusest klastrist c_i
2. kui x_i on klastri ainukene punkt, siis see klaster jääb tühjaks, seega klaster eemaldadakse (ja klastrite arv K on nüüd ühe võrra väiksem)
3. tõmbame uue suuruse c_i proportsionaalselt tõenäosustega:
   a.) olemasoleva klastri jaoks ($k = 1, ..., K$):

$$p(c_i = k) \propto \frac{n_{k,-i}}{n + \alpha - 1} F(\mathbf{x}_i|\boldsymbol{\theta}_k)$$

   b.) uue klastri jaoks:

$$p(c_i = K + 1) \propto \frac{\alpha}{n + \alpha - 1} \int F(\mathbf{x}_i|\boldsymbol{\theta}) dG_0(\boldsymbol{\theta})$$

kus integraali väärtuse leiab Görür and Rasmussen [2010].
4. kui $c_i = K + 1$, siis klastrite arv suureneb ühe võrra, uue klastri parameetrid tuleb genereerida parameetrite järeljaotusest:

$$F(\mathbf{x}_i|\boldsymbol{\theta}) G_0(\boldsymbol{\theta})$$

Uuendame klastrite parameetrid (k=1,...,K korral)
1. Igale klastrile genereerime (tõmbame) uued parameetrid järeljaotusest, mis põhineb eeljaotusel G_0 ja andmepunktidel, mis on klastris k:

$$\prod_{\mathbf{x}_i \in klaster_k} F(\mathbf{x}_i|\boldsymbol{\theta}) G_0(\boldsymbol{\theta})$$

Gaussi-Wisharti järeljaotuse parameetrite kuju leiab Murphy [2007].

Peatükk 3

Tulemused

Implementeerisin kõik eelnevalt kirjeldatud mudelid statistikatarkvaras R. Testisin neid nii simuleeritud kui ka reaalsetel andmetel. Selles peatükis on esitatud vastavad tulemused. Programmide koodi leiab http://www.stat24.ee/bsc/kood.

3.1 Sissejuhatav näide - iiriste andmestik

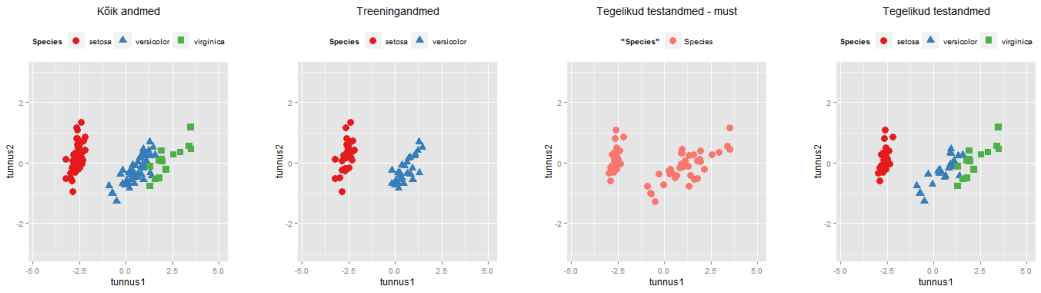
Esmalt kirjeldame mudelite tulemusi iiriste andmestikul. See on mitmemõõtmeline andmestik iirise lillesortidest (*setosa, versicolor, virginica*), kus iga sordi kohta on teada 50 taime kroonlehe ja tupplehe laius ning kõrgus. Andmestiku on koostanud Edgar Anderson ning esmalt kasutas seda Ronald Fisher diskriminantanalüüsil.

Oletame, et bioloogid uurivad iiriseid ning nad teavad, et eksisteerib kaht liiki iiriseid: *setosa* ja *versicolor*. Jagasime andmestiku juhuslikult treening- ja testandmestikuks, kus esimeses pole ühtegi näidist sordist *virginica*, teises neist on aga mitmeid selle sordi esindajaid.

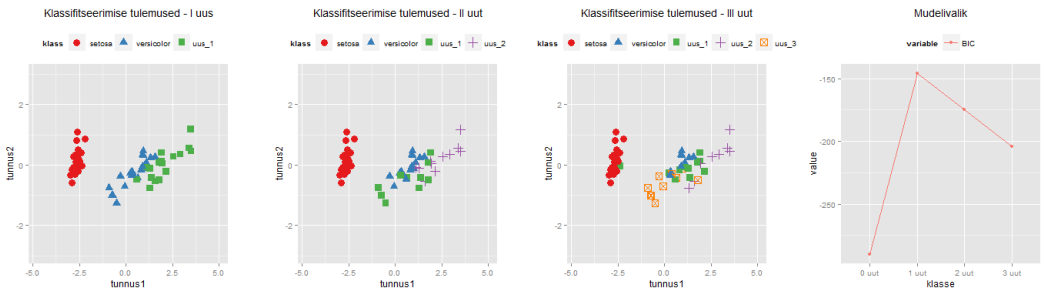
Joonisel 3.1 on esmalt näidatud sildistatud täisandmestik, järgnevalt on toodud sildistatud treeningandmed ehk andmed, mida bioloogid on näinud ja mille sorti nad teavad. Seejärel on näidatud testandmestik, kus bioloogide ülesandeks on igale taimele määrata sort. Selles andmestikus on vaatlusi klassist, mida bioloogid pole minevikus näinud (sordist *virginica*) ja arvatavasti klassifitseeriksid nad need vaatlused kui *setosa* või *versicolor*. Viimasena on näidatud testandmes-

tiku õiged sildid.

Rakendasime eelnevalt kirjeldatud mudeleid sellele andmestikule. Kõik mudelid tuvastasid, et andmestikus on üks uus klass. Joonisel 3.2 on toodud induktiivse mudeli tulemused. Näidatud on klassifitseerimise tulemused, kui andmetele sobitati mudel, mis eeldas vastavalt ühte, kahte või kolme uut klassi. Näeme, et BIC kriteeriumi põhjal valime *parimaks* mudeliks variandi, mis eeldab ühte uut klassi.



Joonis 3.1: Iiriste andmestik. Näidatud on treening- ja testandmed.

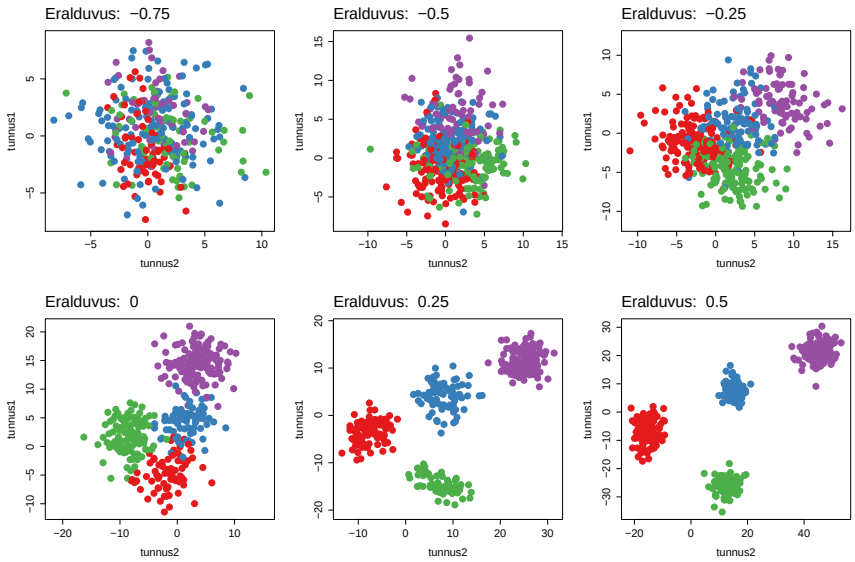


Joonis 3.2: Induktiivne mudeli tulemused. Näidatud on treeningandmetele sobitatud 1, 2 ja 3 uue klassiga mudelite klassifitseerimisotsused. Lisaks on näidatud BIC väärtused.

3.2 Võrdlus genereeritud andmetel

Teises peatükus kirjeldasime kahte EM-algoritmil põhinevat meetodit andmete klassifitseerimiseks ja uute klasside leidmiseks: transduktiivne ja induktiivne mu-

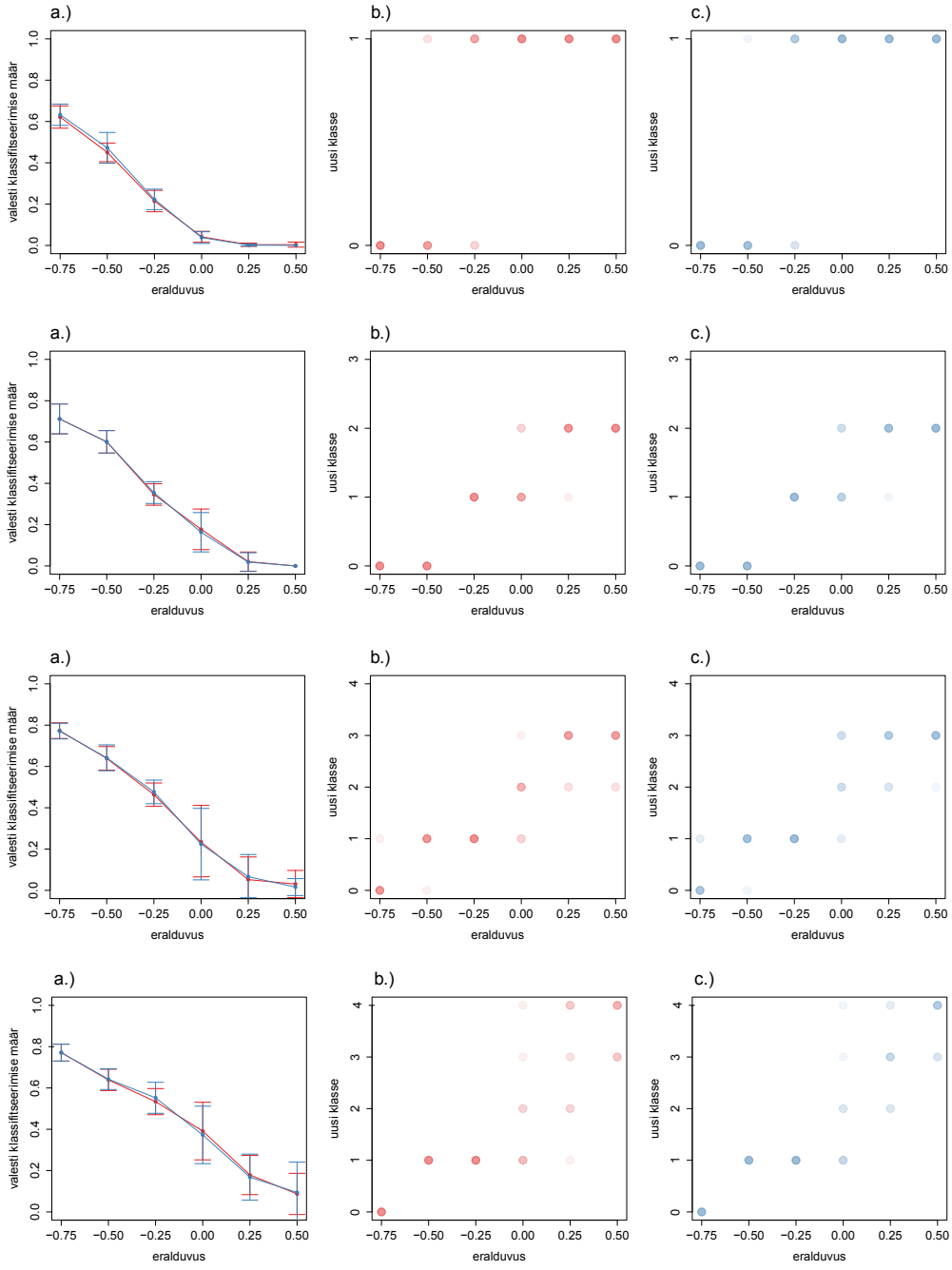
del. Võrdleme nende algoritmide klassifitseerimise ja mudeli valiku headust. Nimelt genereerisime erineva eralduvusega andmestikke, jagasime andmed juhuslikult treening- ja testandmeteks. Seejärel kaotasime treeninandmetest kas 1, 2, 3 või 4 klastri andmed. Erineva eralduvusega klastrite genereerimiseks kasutati Ri paketti *clusterGeneration*, eralduvuse määr on vahemikus $(-1, 1)$ ja täpsema info selle kohta leiab Qiu and Joe [2006]. Erineva eralduvusega andmestikud on illustreeritud joonisel 3.3.



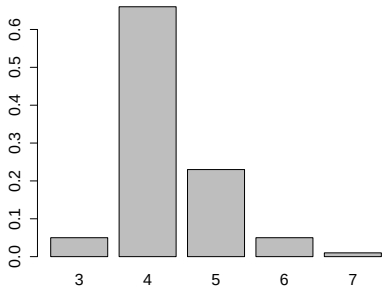
Joonis 3.3: Erineva eraldusmääraga andmestikud.

Kontrollisime klassifitseerimise headust ja seda, kas uued klassid leitakse üles. Joonisel 3.4 on näidatud valesti klassifitseerimise määr ja leitud uute klasside arv erinevate klastrite kattuvuse korral. Kuna tulemustest suuri erinevusi ei ole ning induktiivne mudel on arvutuslikult kiirem, siis eelistatum neist kahest on induktiivne mudel.

Eralduvuse määra -0.25 juures indikeerivad nii induktiivne kui ka transduktiivne mudel, et andmetes on lisaks teadaolevatele kalssidele veel üks klass. Joonisel 3.5 on näidatud sellisel andmestikul treenitud Dirichlet protsessi segumudeli korral saadud klastrite arvu järeljaotus.



Joonis 3.4: Transduktiivse (sinine) ja induktiivse (punane) mudeli võrdlus. Treeningandmetes on 3 erinevat klassi, testandmetes on lisaks 1,2,3 või 4 uut klassi (ridade kaupa).



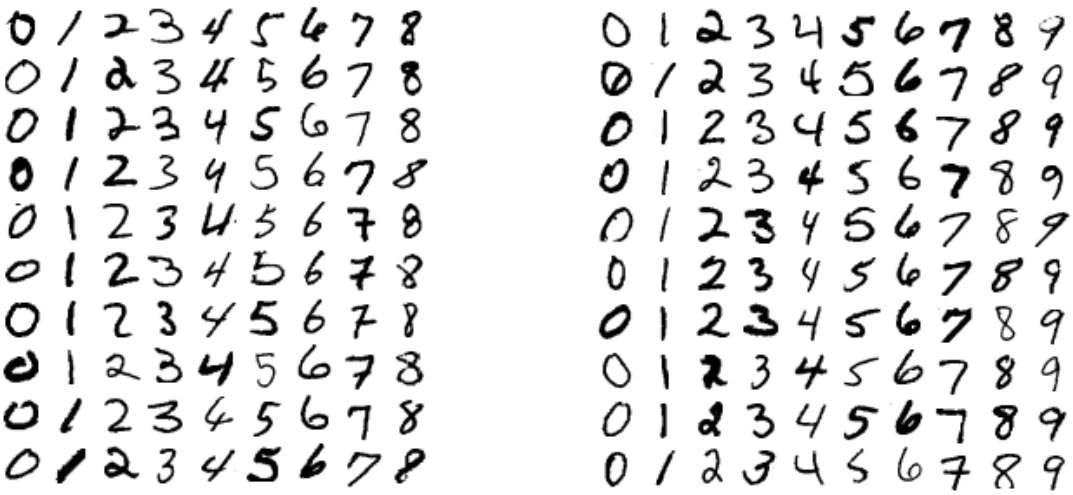
Joonis 3.5: Dirichlet protsessi segumudel - klastrite järeljaotus.

3.3 Numbrite andmestik

Testisime EM-algoritmil põhinevaid mudeleid käsitsi kirjutatud numbrite andmestikul. Iga number on esitatud 28 x 28 pikslisel mustvalgel pildil. Seega on meil ühe pildi korral teada 784 piksli halli tooni intensiivsus, mida iseloomustatakse numbriga vahemikus 0 kuni 255. Et andmestik koonduks mõistliku ajaga, vähendati andmete mõõtmelisust. Selleks kasutati peakomponendite analüüsi ja arvesse võeti 50 esimest komponenti, mis kirjeldasid andmetest 82.6 %.

Jagasime andmestiku juhuslikult treening- ja testandmestikuks, kusjuures esimeses neist ei olnud ühtegi vaatlust numbri 9 kohta, testandmestik sisaldas aga mitmeid vaatluseid number 9-st (joonis 3.6). Klassikalised algoritmid sildistaksid number üheksad kui ühena teadaolevatest klassidest ja ei oleks võimelised uut gruppi tuvastama. Meie algoritm leidis aga lausa 3 uut klassi. Tabel 3.1 kirjeldab induktiivse mudeli tulemusi. Näeme, et teadaolevate klasside klassifitseerimisega saab algoritm väga hästi hakkama. Esimesed kaks uut klassi sisaldavad neljasid, seitsmeid ja üheksaid, kolmas uus klass sisaldab kõiki numbreid ühtlaselt.

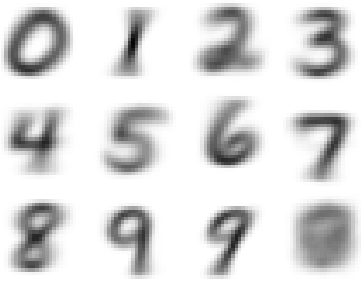
Võib tunduda, et algoritm ebaõnnestus nende andmete klassifitseerimisel, sest ühe number üheksate klassi asemel leiti tervelt kolm uut klassi. Vaadates aga iga klassi keskmise numbri joonist (joonis 3.7) on algoritmi tulemus vägagi mõistlik: leiti püstise ja viltuse jalaga (üheksate) klass, lisaks neile veel müraklass.



Joonis 3.6: Käsitsikirjutatud numbrite andmestik. Vasakul on näidatud valim treeningandmetest, paremal testandmetest. Treeningandmed ei sisalda vaatlusi number üheksast.

Tabel 3.1: Segadusmaatriks induktiivse mudeli klassifitseerimistulemuste kohta käsitsikirjutatud numbrite andmestikul.

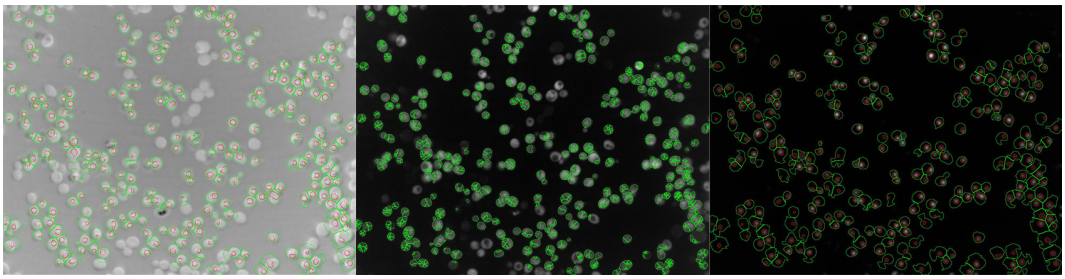
	0	1	2	3	4	5	6	7	8	uus_1	uus_2	uus_3
0	919	0	0	0	0	0	0	0	3	0	0	65
1	0	1080	6	2	2	0	0	1	9	0	1	24
2	4	1	881	1	1	0	0	3	7	0	0	96
3	0	0	8	957	0	10	0	1	9	6	1	70
4	0	0	0	0	771	0	1	2	3	51	85	62
5	2	0	0	6	0	828	4	0	2	2	4	51
6	2	0	0	0	0	13	901	0	5	0	0	47
7	0	2	10	0	0	0	0	644	4	89	223	97
8	2	4	2	5	0	2	0	1	858	0	0	98
9	1	0	1	4	4	2	0	4	13	465	419	116



Joonis 3.7: Induktiivse mudeli tulemus käsitsikirjutatud numbrite andmestikul. Näidatud on iga klastri keskmine number.

3.4 Bioloogilised andmed

EM-algoritmil põhinevaid mudeleid testisime ka bioloogistel andmetel, mis pärinevad prof. Brenda Andrewsi laborist *Donnelly Center for Cellular and Biomolecular Research*. Pärmirakkudega on tehtud teatud eksperimente, mille kohta on olemas pildiandmed (joonis 3.8), mis näitavad, mis osa rakust (näiteks vakuool, tuum, Golgi kompleks vms) helendab. Pildiandmeid on seejärel töödeldud ja saadud üle 300 erineva tunnuse, mis kirjeldavad näiteks helenduse intensiivsust, kuju, tekstuuri, asukohta jms. Sellisel kujul saadud andmeid testisime sarnaselt eelnevatega: jagasime andmestiku juhuslikult kaheks, testandmetest kaotasime ära ühe klassi ja vaatlesime, kas algoritm leiab selle klassi ülesse.



Joonis 3.8: Pildid, mille põhjal on töös testitud bioloogiline andmestik koostatud. Näidatud on kolm laserkanalit eri värvide jaoks samast kaadrist.

Tabelis 3.2 on näidatud induktiivse algoritmi tulemus, kui testandmetest eemaldati klass *vacuole*. Mudeli valik andis tulemuseks ühe uue klassiga mudeli ning

tabelist on näha, et algoritm on klassifitseerimisega hakkama saanud.

Tabel 3.2: Segadusmaatriks induktiivse mudeli klassifitseerimistulemuste kohta bioloogilistel andmetel.

	1	2	3	4	5	6	7	8	9	10	uus_1
bud_neck	11	0	0	1	0	1	0	0	0	3	4
cell_periphery	0	33	0	0	1	5	0	0	0	0	4
cytoplasm	0	0	36	0	5	0	0	0	0	0	0
endosome	0	0	0	28	2	0	0	0	0	1	3
ER	0	2	0	0	15	0	0	0	0	0	1
Golgi_early_Golgi	0	0	0	0	1	38	1	0	0	0	7
nuclear_periphery	0	0	1	0	0	0	26	0	1	1	2
nucleolus	0	0	0	0	0	0	0	76	0	0	2
nucleus	0	0	0	1	0	0	4	0	37	2	0
peroxisome	0	0	0	1	0	0	2	0	1	22	7
vacuole	0	0	0	0	0	4	1	1	0	1	33

Kokkuvõte

Klassikalised diskrimineerimismeetodid eeldavad, et uuritava populatsiooni kõik klassid on esindatud treeningandmetes. Sageli võib see eeldus olla liialt range ning loomulik on soovida, et juhul kui andmetes esineb uusi andmegruppe, leiab klassifitseerija need ka üles. Töös uuriti, implementeeriti ja katsetati meetodeid, mis on võimelised osaliselt sildistatud andmeid nii klassifitseerima kui ka uusi andmegruppe neis tuvastama.

Esiteks kirjeldati kahte meetodit, mis põhinesid Gaussi segumudelil ja EM-algoritmil. Esimene neist - transduktiivne viis - hindas mudeli kõik parameetrid test- ja treeningandmete põhjal korraga. Induktiivse meetodi korral olid teadaolevate klasside parameetrid fikseeritud treeningandmete põhjal ning testandmed võisid mõjutada vaid uute klastrite parameetreid. Uute klasside arvu tuvastamiseks hinnati erineva klastrite arvuga mudelid ning kasutati Bayesi informatsioonikriteeriumi nende hulgast parima mudeli valimiseks.

Genereeritud andmetel tehtud võrdlused näitasid, et nende meetodite tulemused on sarnased. Transduktiivne mudel on aga arvutuslikult mahukam, sest hindab teadaolevate klasside parameetrid alati uuesti. Seega on neist kahest mõistlikum kasutada induktiivset mudelit.

Töös uuriti ka Dirichlet' protsessi segumudelit. See on mitteparameetriline Bayesi segumudel, kus komponentide osakaalude eeljaotuseks on Dirichlet' protsess. Mudel eeldab lõpmatu arvu segukomponente ning väldib mudelivaliku sammu. Mudeli parameetrite hindamiseks kasutati Gibbsi valikut.

Dirichlet' protsessi segumudel on teoreetiliselt elegantne, kuid selle implementeerimine on raske ja parameetrite hindamise algoritmid keerulised. Kasutades Gibbsi valikut parameetrite järeljaotuse leidmiseks, tekib siltide vahetumise probleem, millele pole ilusaid lahendusi. Lisaks on see meetod eelnevatest mudelitest

arvutusmahukam.

Tööd on võimalik mitmeti edasi arendada. Parameetrite hindamiseks võib stohhastilise Gibbsi valiku asemel kasutada deterministlikke meetodeid nagu variatsiooniline Bayes. Sel juhul väldiksime siltide vahetumise probleemi, kuid kaotaksime järeljaotuse hinnangu täpsuses. Lisaks on Dirichlet' protsessi segumudeli korral võimalik vältida parameetri α valimist, kui muuta mudel hierarhiliseks ehk anda parameetrile α eeljaotus.

Kasutajale, keda huvitab kiire ja mõistlik tulemus, sobib kõige paremini induktiivne mudel, kus teadaolevate klasside parameetrid õpitakse vaid üks kord. Nägime, et mudel töötas reaalsetel andmestikel hästi: ta sai hakkama nii bioloogiliste andmete kui ka käsitsikirjutatud numbrite andmestiku klassifitseerimise ja uute klasside tuvastamisega. Tihti on aega mõistlikum kulutada andmetest arusaamisele veidi ebatäpsema kuid lihtsama mudeli abil, kui üritada sobitada keerulisemat ja aeganõudvat ideaalset mudelit.

Semi-supervised learning of mixture models

Bachelor's Thesis

Tanel Pärnamaa

Summary

An important problem usually not taken into account in classification is the possibility that test data may have classes that are not observed in the training phase. Classical methods classify those data points into one of the known classes in the training set and are not able to detect novel classes. The aim of this thesis is to describe, implement and test models that are able to detect several novel classes of points.

Firstly, we described two algorithms to learn a mixture model with novel classes. Expectation maximization algorithm was used for parameter estimation and Bayesian information criterion was used for selecting the actual number of clusters.

Secondly, we used a Dirichlet process Gaussian mixtures, a nonparametric alternative to standard mixture models. This model assumes infinite number of mixture components and therefore avoids the problem of selecting the right number of clusters. The inference was implemented using Gibbs sampling and conjugate prior distributions.

All of the models were implemented in R and tested on real and artificial data. All models were able to capture the right number of components in the data if the

separating signal was strong. The added flexibility and mathematical aesthetics of the Dirichlet process mixtures were countered by the lack of a closed form solution and the additional computational burden for inference using sampling.

Viited

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. ISBN 0387310738. 7, 8, 11

D. Blackwell and J. B. MacQueen. Ferguson distributions via polya urn schemes. *Annals of Statistics*, 1:353–355, 1973. 24

D. Blei. Bayesian nonparametrics. University Lecture, 2007. 26

C. Bouveyron. Adaptive mixture discriminant analysis for supervised learning with unobserved classes. 2010. 12

B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the dirichlet distribution and related processes. 2010. 16, 20

D. Görür. Nonparametric bayesian discrete latent variable models for unsupervised learning. *PhD thesis*, 2007. 16

D. Görür and C. E. Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25:615–626, 2010. 31, 34

M. R. Gupta and Y. Chen. Theory and use of the em algorithm. *Signal Processing*, page 223–296, 2010. 11

K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. 2007. 34

R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 2006. 16

W. Qiu and H. Joe. Separation index and partial membership for clustering. *Computational Statistics and Data Analysis*, 50:585–603, 2006. 37

A. E. Raftery. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1995. 32

J.Šethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4: 639–650, 1994. 21

E. B. Sudderth. Graphical models for visual object recognition and tracking. *PhD thesis*, 2006. 16

Y. W. Teh. Dirichlet process. 2010. 16

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 2005. 16

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Tanel Pärnamaa (sünnikuupäev 8. septmeber 1991),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **"Segumudeli õppimine osaliselt sildistatud andmetest"**, mille juhendajad on Leopold Parts ja Raivo Kolde,

   (a) reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

   (b) üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus 6. mail 2013