Antal Péter, Antos András, Horváth Gábor, Hullám Gábor, Kocsis Imre,
Marx Péter, Millinghoffer András, Pataricza András, Salánki Ágnes

Intelligens adatelemzés

Szerkesztette: Antal Péter

A jegyzetben az adatelemzés folyamata szerint ismertetünk „intelligens” megközelítéseket, amelyekben az intelligencia a felhasznált háttértudás, a számítási erőforrások vagy a modellek komplex volta miatt is jelenik meg.

Budapesti Műszaki és Gazdaságtudományi Egyetem

MŰEGYETEM 1782

Semmelweis Egyetem

ÚJ SZÉCHENYI TERV

Typotex Kiadó 2014

www.interkonyv.hu

Lektorálta: Takács Gábor
Készült a Typotex Kiadó gondozásában
Felelős vezető: Votisky Zsuzsa

ISBN 978-963-279-171-5
Készült a TÁMOP-4.1.2/A/1-11/1-2011-0079 számú, „Konzorcium a biotechnológia és bioinformatika aktív tanulásáért” című projekt keretében.

Nemzeti Fejlesztési Ügynökség
www.ujszechenyiterv.gov.hu
06 40 638 638

MAGYARORSZÁG MEGÚJUL

A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Tartalomjegyzék

1. fejezet

Aktív tanulás

1.1. Bevezetés: becslés, döntés, felügyelt (passzív) tanulás

Röviden összefoglaljuk a becslések, döntések és felügyelt (supervised) tanulás leírására használt terminológiát (ld. a [79] jegyzet 5. fejezet, a [35] könyv 1. és 2. fejezet, a [59] könyv 1. fejezet):

Ha egy X megfigyelhető mennyiségből kell egy másik, (még) nem megfigyelhető Y mennyiségre következtetnünk, akkor X-et és Y-t valószínűségi változókkal modellezhetjük. Ezek értékkészletét jelölje rendre $\mathcal{X}$ és $\mathcal{Y}$. $\mathcal{X}$ és $\mathcal{Y}$ lehet pl. $\mathbb{R}$ (a valós számok halmaza), $\mathbb{R}^d$, $\{0,1\}^d$, $[0,1]^d$ vagy ezeknek egy megszámlálható, esetleg véges részhalmaza. $\mathcal{Y}$ elemeit *címkéknek* is nevezzük.

X értékéből Y-t a $g : \mathcal{X} \to \bar{\mathcal{Y}}$ függvénnyel próbáljuk meghatározni ($\bar{\mathcal{Y}}(\supseteq \mathcal{Y})$ a szóba jövő függvények értékkészletének uniója). A $g(X)$ *következtetés* jóságát egy $C : \mathcal{Y} \times \bar{\mathcal{Y}} \to [0,\infty)$ *költségfüggvény* méri, $C(y, g(x))$ a költsége annak, ha a valódi $Y = y$ érték helyett g az $X = x$ megfigyelésekor $g(x)$-re következtet. Minél kisebb ez a költség, annál jobb $g(x)$. Ha a költségfüggvény értéke pontatlan következtetés esetén nem függ a következtetés konkrét értékétől, azaz minden $y \in \mathcal{Y}$-ra $C(y,\cdot)$ az $\bar{\mathcal{Y}} \setminus \{y\}$-on konstans, akkor *döntési problémáról* beszélünk. Itt általában $\bar{\mathcal{Y}} = \mathcal{Y}$ diszkrét (megszámlálható), pl. $\mathcal{Y} = \{0,1\}$. Ha C valamiféle intuitív távolság, akkor *becslési problémáról* beszélünk. Itt általában $\bar{\mathcal{Y}}$ vagy akár $\mathcal{Y}$ is folytonos.

g jóságát várható költsége, az $R(g) \stackrel{\text{def}}{=} \mathbf{E}\left[C(Y, g(X))\right]$ *globális kockázat* méri. Azokat a g-ket, amelyekre $R(g)$ a legkisebb, *optimálisnak* nevezzük.

Legyen $r(g,x) \stackrel{\text{def}}{=} \mathbf{E}\left[C(Y, g(X))|X = x\right]$ a *lokális kockázat függvény.*

Nyilván $\mathbf{E}\left[r(g,X)\right] = R(g)$ és

$$r(g,x) = \mathbf{E}\left[C(Y,g(x))|X=x\right] = \int_{\mathcal{Y}} C(y,g(x))\, dF_{Y|x}(y), \tag{1.1}$$

ahol $F_{Y|x}$ az Y feltételes, ún. *a posteriori eloszlásfüggvénye*, ha $X = x$. A fenti általános integrál a továbbiakban általunk vizsgált esetekben diszkrét, illetve abszolút folytonos

eloszlások esetén egyszerű összegzésre, illetve Riemann-integrálra vezet.

Döntési problémáknál a leggyakoribb költségfüggvény választás a $C_0(y,y') \stackrel{\text{def}}{=} \mathbf{I}_{\{y\neq y'\}}$ *0-1* költség (ahol $\mathbf{I}_A$ az A esemény indikátorfüggvénye, azaz $\mathbf{I}_A = 1$, ha A bekövetkezik és $\mathbf{I}_A = 0$, ha nem). Ekkor $R(g) = \mathbf{P}[Y \neq g(X)]$, ami g hibázásának a valószínűsége.

Becslési problémáknál $\bar{\mathcal{Y}} = \mathcal{Y} = \mathbb{R}^d$-re pedig gyakori választás az L_2 távolság ($\|\cdot\|$ norma) négyzete, azaz a $C_2(y,y') \stackrel{\text{def}}{=} \sum_{s=1}^{d} |y_s - y'_s|^2 = \|y-y'\|^2$ *négyzetes* költség. Ekkor pl. $\bar{\mathcal{Y}} = \mathbb{R}$-re $R(g) = \mathbf{E}[(Y-g(X))^2]$ a *négyzetes középhiba.*

1.1.1. Bayes-döntés

Vizsgáljuk meg először alaposabban azt az esetet, amikor $\mathcal{Y} = \bar{\mathcal{Y}}$ véges, most a tömörség kedvéért a $\{0,1\}$ halmaz és a költség $C(i,j) = C_0(i,j) = \mathbf{I}_{\{i\neq j\}}$. Ekkor

1.1.1. Definíció. $i \in \{0,1\}$-re az $\{Y = i\}$ eseményt *i-dik hipotézisnek* nevezzük. Y a posteriori eloszlását az $\eta_i(x) \stackrel{\text{def}}{=} \mathbf{P}[Y = i | X = x]$ *a posteriori valószínűségek* adják meg. g-t *döntésfüggvénynek* nevezzük. 0 és 1 g-vel való ősképei $\mathcal{X}$ egy partícióját adják, amelynek $D_i = \{x \in \mathcal{X} : g(x) = i\}$ osztályait *döntési tartományoknak* nevezzük.

A (D_0, D_1) döntési tartományok teljesen meghatározzák a g döntésfüggvényt, azaz a döntésfüggvényt megadhatjuk a döntési tartományok rendezett párjával. g és (D_0, D_1) között a kapcsolat nevezetesen $\{g(x) = j\} \Leftrightarrow \{x \in D_j\}$. Most (1.1) szerint a lokális kockázat

$$r(g,x) = \mathbf{I}_{\{g(x)=1\}}\eta_0(x) + \mathbf{I}_{\{g(x)=0\}}\eta_1(x) = 1 - \mathbf{I}_{\{x\in D_0\}}\eta_0(x) - \mathbf{I}_{\{x\in D_1\}}\eta_1(x).$$

Látható, hogy a fenti $r(g,x)$-et olyan g minimalizálja, amely x-et a nagyobb $\eta_j(x)$-hez tartozó D_j tartományba sorolja (ld. 1.1. tétel alább).

Legyenek tehát a D_j^* tartományok olyanok, hogy $\forall x \in D_j^*$ akkor és csak akkor, ha $\eta_j(x) > \eta_{1-j}(x)$ vagy $j = 0$ és $\eta_0(x) = \eta_1(x)$. Tehát g^* azt a hipotézist választja, amelyik a valószínűbb a megfigyelés ismeretében, azaz g^*-ot a maximális a posteriori valószínűségek megkeresésével határozhatjuk meg. D_0^* és D_1^* az $\mathcal{X}$ partícióját adják, továbbá

$$x \in D_j^* \Rightarrow \eta_j(x) = \max(\eta_0(x), \eta_1(x)). \tag{1.2}$$

1.1.2. Definíció. A fenti (D_0^*, D_1^*) tartományok által meghatározott g^* döntésfüggvényt (azaz amire $g^*(x) = j \Leftrightarrow x \in D_j^*$) *Bayes-döntésnek*, vagy *maximum a posteriori döntésnek* nevezzük.

1.1. Theorem. *A Bayes-döntés $\forall x$-re minimalizálja a lokális kockázatot, és így optimális. A minimum értéke $r(g^*,x) = \min(\eta_0(x), \eta_1(x))$.*

Tehát a Bayes-döntés (optimális) globális kockázata

$$R(g^*) = \mathbf{E}[r(g^*,X)] = \mathbf{E}[\min(\eta_0(X), \eta_1(X))] = \mathbf{E}[\min(\eta_1(X), 1-\eta_1(X))].$$

Ezt *Bayes-kockázatnak* vagy *Bayes-hibának* is nevezzük. [35, 2.1. fejezet]

1.1.2. Bayes-döntés közelítése

Az $\eta_i(x)$-k pontos értékei általában ismeretlenek. Tekintsünk egy ilyen szituációt, amelyben azonban a η_i-ket meg tudjuk becsülni valamely $\tilde{\eta}_i$ függvényekkel. Az (1.2)-et teljesítő Bayes-döntés mintájára a $(\tilde{\eta}_0, \tilde{\eta}_1)$ függvényekhez is hozzárendelhetünk analóg módon olyan $\tilde{g}$ döntésfüggvényt, amely x esetén a nagyobb $\tilde{\eta}_j(x)$-hez tartozó j-re dönt, azaz

$$\tilde{g}(x) = j \Rightarrow \tilde{\eta}_j(x) = \max(\tilde{\eta}_0(x), \tilde{\eta}_1(x)). \tag{1.3}$$

(Egyenlőség esetén tetszőleges elv szerint választunk.) $\tilde{g}$ tehát úgy viszonyul az $\tilde{\eta}_i$-khez, ahogy g^* az η_i-khez. A következő tétel szerint ha az $\tilde{\eta}_i$-k jó becslések, akkor $\tilde{g}$ hibája közel lesz g^* hibájához (kisebb természetesen nem lehet az 1.1. tétel alapján), azaz a szóban forgó kockázatok különbsége korlátozható az $\tilde{\eta}_i$-k becslési hibájával:

1.2. Theorem. *$i \in \{0,1\}$-re legyen $\tilde{\eta}_i : \mathcal{X} \to [0,1]$ az η_i becslése és $\tilde{g}$ egy a $(\tilde{\eta}_0, \tilde{\eta}_1)$-hez rendelt döntésfüggvény. Ekkor*

$$r(\tilde{g}, x) - r(g^*, x) \leq \mathbf{I}_{\{\tilde{g}(x) \neq g^*(x)\}}(|\tilde{\eta}_0(x) - \eta_0(x)| + |\tilde{\eta}_1(x) - \eta_1(x)|)$$

és

$$R(\tilde{g}) - R(g^*) \leq \mathbf{E}\left[\mathbf{I}_{\{\tilde{g}(X) \neq g^*(X)\}} \sum_{i \in \{0,1\}} |\tilde{\eta}_i(x) - \eta_i(x)|\right] \leq \mathbf{E}\left[\sum_{i \in \{0,1\}} |\tilde{\eta}_i(X) - \eta_i(X)|\right]. \tag{1.4}$$

Ha feltesszük, hogy $(\tilde{\eta}_0(x), \tilde{\eta}_1(x))$ minden $x \in \mathcal{X}$-re eloszlást alkot, azaz $\tilde{\eta}_0(x) = 1 - \tilde{\eta}_1(x)$, akkor ezek tovább egyszerűsödnek a következőképpen

$$r(\tilde{g}, x) - r(g^*, x) \leq 2\mathbf{I}_{\{\tilde{g}(x) \neq g^*(x)\}} |\tilde{\eta}_1(x) - \eta_1(x)|$$

és

$$R(\tilde{g}) - R(g^*) \leq 2\mathbf{E}\left[\mathbf{I}_{\{\tilde{g}(X) \neq g^*(X)\}} |\tilde{\eta}_1(X) - \eta_1(X)|\right] \leq 2\mathbf{E}\left[|\tilde{\eta}_1(X) - \eta_1(X)|\right].$$

1. példa. Mutassuk meg, hogy $\eta_1(X) = \mathbf{E}[Y|X]$. Tehát ha van egy jó becslésünk Y feltételes várható érték függvényére, akkor az ahhoz (1.3) alapján rendelt döntésfüggvény közel optimális.

Osztályozás mintákból

Tegyük fel, hogy az $\eta_i(x)$-k ismeretlenek, azonban rendelkezésre áll n számú, az (X, Y) párral azonos eloszlású, független minta, azaz az (X_1, Y_1), (X_2, Y_2),..., (X_n, Y_n) valószínűségi változó párok. Ezek alapján próbáljuk közelíteni (X, Y) együttes eloszlását, az $\eta_i(x)$-ket, majd a Bayes-döntést. Ezt a problémát *osztályozásnak* (vagy *alakfelismerésnek*, bizonyos esetben *(felügyelt) tanulásnak*) is hívják.

Legyen most $\mathcal{X}$ is véges, $r_i(x) = \mathbf{P}[X = x, Y = i]$ $(x \in \mathcal{X}, i \in \{0,1\})$ és $\tilde{r}_{in}(x)$ az $r_i(x)$ becslése az n mintából. Mivel $\eta_i(x) = r_i(x)/\mathbf{P}[X = x]$, legyen $\eta_i(x)$ közelítése

$$\tilde{\eta}_{in}(x) = \frac{\tilde{r}_{in}(x)}{\mathbf{P}[X = x]}, \qquad i \in \{0,1\}$$

és $\tilde{g}_n$ a (1.3) alapján hozzájuk rendelt döntésfüggvény. Vegyük észre, hogy $\tilde{\eta}_{in}(x)$-k valójában nem igazi meghatározható becslések, hiszen a nevezőben a $\mathbf{P}[X = x]$ valószínűségeket nem ismerjük, azonban nincs is rájuk szükségünk $\tilde{g}_n$ meghatározásához, hiszen a maximális $\tilde{\eta}_{in}(x)$-t – és így $\tilde{g}_n(x)$ értékét – az a hipotézis adja, amelyikre $\tilde{r}_{in}(x)$ maximális.

A 1.2. tétel (1.4) alakja szerint

$$\begin{aligned} R(\tilde{g}_n) - R(g^*) &\leq \mathbf{E}\left[\sum_{i\in\{0,1\}} |\tilde{\eta}_{in}(X) - \eta_i(X)|\right] = \sum_{x\in\mathcal{X}} \sum_{i\in\{0,1\}} |\tilde{\eta}_{in}(x) - \eta_i(x)|\mathbf{P}[X = x] \\ &= \sum_{x\in\mathcal{X}} \sum_{i\in\{0,1\}} |\tilde{r}_{in}(x) - r_i(x)|. \end{aligned} \tag{1.5}$$

Becsüljük $r_i(x)$-t a megfelelő mintabeli relatív gyakoriságokkal (amely a valószínűségek természetes becslése), azaz legyen $\tilde{r}_{in}(x) = \frac{1}{n}\sum_{t=1}^{n} \mathbf{I}_{\{X_t=x, Y_t=i\}}$. A nagy számok erős törvénye szerint amint $n \to \infty$, a relatív gyakoriságok 1-valószínűséggel tartanak a megfelelő valószínűségekhez, azaz $\forall x \in \mathcal{X}$,$i \in \{0,1\}$, $\tilde{r}_{in}(x) \to r_i(x)$. Ezért (1.5) jobb oldala 0-hoz tart 1-valószínűséggel, és mivel a bal oldalán $R(\tilde{g}_n) - R(g^*)$ nemnegatív, az is. Így eljárásunkkal a Bayes-hibát „majdnem biztosan" tetszőlegesen megközelítő (amint $n \to \infty$) hibájú döntésfüggvény-sorozatot kaptunk. Az osztályozásban ezt úgy is mondják, hogy véges $\mathcal{X}$-ek esetén ez az eljárás *erősen konzisztens*.

1. Remark. (1.5) *jobb oldalának a várható értékét elemezve kiderül, hogy ha a minták száma összemérhető az (X, Y) pár effektív értékkészletével (vagy kisebb), akkor bár az egyes tagok kicsik, az összeg már nem lesz az, így a kapott döntésfüggvény hibavalószínűsége megbízhatatlanná válik.*

1.1.3. Bayes-becslés

Szemben az eddigi 1.1.1–1.1.2. fejezettel, ahol $\mathcal{Y} = \bar{\mathcal{Y}}$ véges volt és a 0-1 költséget vizsgáltuk, azaz Y értékét el akartuk *dönteni*, az alábbi fejezetekben $\mathcal{Y}$ és $\bar{\mathcal{Y}}$ $\mathbb{R}$ többnyire végtelen, sőt folytonos részhalmaza, és C valamiféle távolságát méri a valódi és a *becsült* paraméternek, tehát a hibázás nagysága is számít. Ekkor g-t *becslésfüggvénynek* is nevezzük. A probléma továbbra is az $R(g)$-t minimalizáló $g : \mathcal{X} \to \mathcal{Y}$ megtalálása.

1.1.3. Definíció. *Bayes-becslésnek* nevezzük az optimális g^* becslésfüggvényeket, azaz amikre $R(g^*) = \min_g R(g)$.

Fennáll a következő elégséges feltétel:

1.3. Theorem. *Ha egy g^* becslésfüggvény lokális kockázatára*

$$r(g^*, x) = \min_{y \in \bar{\mathcal{Y}}} \mathbf{E}\left[C(Y, y)|X = x\right]$$

minden $x \in \mathcal{X}$-re, akkor g^* Bayes-becslés.

1.1.4. Regressziós becslés; négyzetes középhiba minimalizálás

Legyen $Y \in \mathbb{R}$ véges szórású változó, és vizsgáljuk a $C(y, y') = C_2(y, y') = (y - y')^2$ négyzetes költséget, azaz keressük azt a g^*-t, amelyre $R(g^*) = \mathbf{E}\left[(g^*(X) - Y)^2\right]$ minimális! A 1.3. tétel szerint, ha $r(g^*, x) = \min_{y \in \bar{\mathcal{Y}}} \mathbf{E}\left[(Y - y)^2|X = x\right]$ minden $x \in \mathcal{X}$-re, akkor g^* Bayes-becslés. Definiáljuk a *regressziós függvény* fogalmát:

1.1.4. Definíció. Regressziós függvénynek nevezzük a $\mu(x) = \mathbf{E}\left[Y|X = x\right]$ függvényt, amely minden x-re Y-nak a feltételes várható értékét adja, ha $X = x$.

1.4. Theorem. *$C = C_2$ esetén bármely g-re $r(g, x) = r(\mu, x) + (\mu(x) - g(x))^2$ $(\forall x \in \mathcal{X})$ és $R(g) = R(\mu) + \mathbf{E}\left[(\mu(X) - g(X))^2\right]$. Így $g^*(X) = \mu(X)$ 1 valószínűséggel, vagyis a Bayes-becslés éppen a regressziós függvény. [59, p. 2]*

1.2. Aktív tanulás fogalma

Míg az eddigi felügyelt tanulásnál a megfigyelések mind címkézve voltak, sok mai praktikus probléma esetén címkézetlen minta bőségesen és olcsón áll rendelkezésre, de azokat igen költséges címkével ellátni (pl. weboldalak, kép-, hang-, videóminták). A *félig felügyelt tanulás* területe azt kutatja, hogyan lehet kihasználni a címkézetlen mintákban rejlő információt. Ha pedig módunk van megválasztani, hogy a minták melyik részéhez kérjünk címkét, *aktív tanulásról* beszélünk [23, 29]. Szemben tehát a szokásos passzív modellekkel, ahol a tanuláshoz a címkéket a tanuló algoritmustól függetlenül kapjuk, az aktív tanulásnál a tanuló (ágens) interaktívan választhatja ki, hogy melyik adatpontokhoz akar címkét. Ettől azt remélhetjük, hogy jelentősen csökkentheti a szükséges címkék számát, ezzel megvalósíthatóbbá téve a problémák gépi tanulás révén való megoldását. Mind empirikusan, mind elméletileg ez a remény bizonyos esetekben indokoltnak tűnik. Akik tanuló algoritmusokat fejlesztenek vagy használnak, gyakran szembesülnek azzal az igénnyel, hogy több címkére lenne szükség. Ilyenkor az aktív tanulás segíthet.

2. példa. Aktív tanulás néhány modern alkalmazása:

1. Gyógyszerkutatás [118]: Adott vegyületek egy óriási (részben virtuális) gyűjteménye, amelyek egy hatalmas dimenziós térben írhatóak le, és közülük olyanokat akarunk találni, amelyek kötődnek egy adott molekulához. Itt a címkézetlen adat a vegyület leírása, a címke, hogy kötődik-e, és a címke nyerése a kémiai kísérlettel történik.
2. Utcai gyalogos detektálás gépjármű kamerával [1]: Fel kell ismerni a gyalogosokat képeken. Itt a címkézetlen pont a kép gyanús téglalapja, címke, hogy gyalogos-e, és a címke nyerése emberi osztályozással történik.

1.3. Véges sok középérték aktív tanulása

Az alábbiakban az egyszerűség kedvéért az 1.1.4. fejezetbeli regressziós becslést ($C = C_2$) vizsgáljuk abban az esetben, amikor $\mathcal{X} = \{1, \dots, K\}$ véges, $Y \in [0,1]$ tetszőleges, Y a posteriori valószínűségei ismeretlenek, azonban g meghatározásához kaphatunk összesen n darab független mintát, az $\{(X_t, Y_t)\}_{1 \le t \le n}$ párokat, amelyekben X_t-t – az 1.1.2. fejezettel ellentétben – aktívan határozhatjuk meg, míg az $Y_t|X_t$ feltételes eloszlása megegyezik az $Y|X$-évek. Ha a tömörség kedvéért X eloszlását $p_k \stackrel{\text{def}}{=} \mathbf{P}[X = k]$, $\mu(k)$-t μ_k és $g(k)$-t $\hat{\mu}_{kn}$ jelöli, akkor 1.4. tétel szerint

$$r(\hat{\mu}_n, k) = r(\mu, k) + (\hat{\mu}_{kn} - \mu_k)^2 \quad (\forall k \in \mathcal{X}) \qquad \text{és} \qquad R(\hat{\mu}_n) = R(\mu) + \sum_{k=1}^{K} p_k (\hat{\mu}_{kn} - \mu_k)^2.$$

A cél tehát az utóbbi tagokat minimalizáló $\hat{\mu}_{kn}$-k találása, azaz $\mathbf{E}[r(\hat{\mu}_n, k)]$ és $\mathbf{E}[R(\hat{\mu}_n)]$ minimalizálandó tagja

$$L_{kn} \stackrel{\text{def}}{=} \mathbf{E}\left[(\hat{\mu}_{kn} - \mu_k)^2\right], \qquad \text{illetve} \qquad L_n^{(p)} \stackrel{\text{def}}{=} \sum_{k=1}^{K} p_k L_{kn}.$$

Ezt úgy is interpretálhatjuk, hogy a μ_k (feltételes) várható értékeket kell becsülni mintákból a becslések L_{kn} négyzetes középhibáinak előírt (és esetleg ismert) $p = \{p_k\}_{k \in \mathcal{X}}$ súlyozása mellett. Több „zajos” mennyiség várható értékének előírt súlyozás által meghatározott (vagy egyforma) pontossággal való becslése igen alapvető probléma. Illusztrációképpen tekintsük pl. azt, hogy egy olyan termék minőségét kell vizsgálni, amely K-féle eltérő körülmények között (*opciók*, *állapotok*) is működtethető, és minden egyes állapotban a terméket tesztelni kell, hogy az elvárt módon viselkedik-e. Egy Y_t mérés a k. állapotban (amikor $X_t = k$) tekinthető egy zajos jel előállításának úgy, hogy a jel μ_k várható értéke megmutatja az eltérést a helyes viselkedéstől. Mivel a mérések eredményei véletlenszerűek, minden egyes opcióhoz több mérés szükséges. Ha egy mérés drága (pl. ha az a termék károsodásával járhat), akkor minden állapotban ugyanannyit mérni gyakran pazarló lehet, hiszen bármely opcióra a becslési pontatlanság arányos lesz az adott opció mérési eredményeinek szórásával, és így, ha az egyik opcióra az eredmények nagy szórásúak, akkor ezt gyakrabban mérhetjük annak árán, hogy a kisebb szórású állapotokban ritkábban mérünk, ezzel kiegyenlítve a becslések minőségét. Ez alapján azt gyaníthatjuk, hogy egy – a méréseknek az opciókhoz rendelésére vonatkozó – jó algoritmus jelentős költségmegtakarítást eredményezhet ahhoz képest, mintha minden egyes opcióra egyforma gyakorisággal mérnénk.

Ha a tanulási folyamat után valamely $\{1, 2, \dots, K\}$ fölötti p eloszlás szerint lesz kiválasztva az a k. opció, amelyhez tartozó μ_k-t kell végül (négyzetes hibában) megbecsülni, akkor a cél az, hogy meghatározzuk a μ_k-kat a p_k-kal súlyozott pontossággal, azaz éppen a fenti $L_n^{(p)}$ minimalizálása.

Ha a tanulás során p nem ismert, akkor a hibát – pesszimista megközelítést alkalmazva

– minden esetben a legrosszabb súlyozásra kell minimalizálnunk, azaz a cél

$$L_n = \max_p L_n^{(p)} = \max_p \sum_{k=1}^{K} p_k L_{kn} = \max_{1\le k\le K} L_{kn}$$

minimalizálása. Ez azt a kívánalmat fejezi ki, hogy mindegyik opció becslése egyformán fontos, amit motiválhat pl. az a lehetséges cél, hogy meghatározzuk a μ_k-kat *ugyanolyan* pontossággal az állapotok teljes tartományán a célból, hogy előállítsuk a termék minőségének *egyenletesen* pontos jellemzését. Az alábbiakban az aktív tanulás ezen esetét tekintjük át.

2. Remark. *A probléma egy további változata merül fel a* rétegzett mintavétel *területén, amely egy fontos szóráscsökkentő technika a Monte-Carlo becslések területén [43]. Ekkor a populáció teljes tartománya ún.* rétegekre *van particionálva. A feladat a* $\mu = \sum_{k=1}^{K} w_k \mu_k$ *várható érték becslése, ahol* μ_k *a* k*. rétegbeli várható érték,* $\{w_k\}$ *pedig a rétegek súlyait meghatátozó adott eloszlás. Ismét bármely adott időpontban csak egy opciót lehet tesztelni (ez megfelel egy adott rétegben való megfigyelés nyerésének). Legyen* $\hat{\mu}_n$ *valamely algoritmus által előállított* n *megfigyelésen alapuló becslés.* $\hat{\mu}_n$ *hibáját az* $L_n^{(w)} = \mathbf{E}\left[(\hat{\mu}_n - \mu)^2\right]$ *négyzetes középhiba méri. Feltehető, hogy a különböző eloszlásokból jövő megfigyelések függetlenek egymástól. Ilyen helyzetben* μ*-t gyakran úgy becsülik, hogy először az egyes opciók várható értékét becsülik meg, majd kiszámítják* $\hat{\mu}_n = \sum_{k=1}^{K} w_k \hat{\mu}_{kn}$*-t, ahol* $\hat{\mu}_{kn}$ *a* μ_k *egy becslése. A különböző opciókból származó adatok függetlensége miatt feltehető, hogy bármely réteg várható értékét (csak) a belőle rendelkezésre álló adatok alapján érdemes becsülni. Ekkor*

$$L_n^{(w)} = \mathbf{E}\left[\Big(\sum_{k=1}^{K} w_k(\hat{\mu}_{kn} - \mu_k)\Big)^2\right] = \sum_{k=1}^{K} w_k^2 L_{kn} + \sum_{1\le k\ne k'\le K} w_k w_k \mathbf{E}\left[(\hat{\mu}_{kn} - \mu_k)(\hat{\mu}_{k'n} - \mu_{k'})\right].$$

A cél ismét az, hogy ez a lehető legkisebb legyen. Determinisztikus allokáció esetén (a különböző rétegekhez tartozó mintaátlagok függetlensége miatt) a második, kovariancia jellegű tag eltűnik, így ekkor $L_n^{(w)} = \sum_{k=1}^{K} w_k^2 L_{kn}$*. Látható, hogy ezen veszteség minimalizálása szinte azonos az előzővel, csak most a veszteségben* p_k *helyett* w_k^2*-tel súlyozunk [17]. Ha az allokáció függ a korábbi mintáktól (ld. alább), a kovariancia tag nem mindig nulla, de általánosabb esetekben is többnyire kicsi.*

Bármely $1 \le k \le K$-ra tehát a döntéshozó eldöntheti, hogy hányszor választja $X_t = k$-t, azaz hányszor generál az $Y|\{X = k\}$ feltételes eloszlásból független mintákat. (Ezek megfelelnek a fenti példában a vizsgálatok eredményeinek.) A k. opcióról t-edszerre kért minta értékét jelöljük Y_{kt}-vel is. Hogy melyik opcióhoz hány mintát *allokáljunk*, az szorosan kapcsolódik a statisztikai *optimális kísérlettervezéshez* is [42, 19].

Azonban most a megfigyeléseket szekvenciálisan végezhetjük: a t. tesztig gyűjtött információk alapján döntheti el a döntéshozó, hogy melyik opciót válassza legközelebb. Ez, és az, hogy egyszerre csak egy opciót (*kart*) lehet tesztelni, hasonlít a *többkarú rabló problémákhoz* [75, 8] is. Ugyanakkor a teljesítőképességet mérő kritérium más, mint a rablóproblémánál használt, ahol a megfigyelt értékeket tekintjük nyereségnek és a tanulás alatti

teljesítőképesség számít. Ugyanakkor látni fogjuk, hogy a klasszikus rabló-problémára jellemző *felfedezés–kihasználás dilemma* itt is jelentős szerepet játszik.

Mivel μ_k várható érték, itt most azt feltételezzük, hogy a $\hat{\mu}_{kn}$ becslés a k. opció mintaátlagaként áll elő:

$$\hat{\mu}_{kn} = \frac{1}{T_{kn}} \sum_{t=1}^{T_{kn}} Y_{kt},$$

ahol $T_{kn} = \sum_{t=1}^{n} \mathbf{I}_{\{X_t=k\}}$ jelöli, hogy hányszor kértünk megfigyelést a k. opcióról az n. mintáig.

Tekintsük a probléma nem szekvenciális változatát, vagyis $T_{1n}, \ldots, T_{Kn}$ előzetes megválasztásának kérdését úgy, hogy $T_{1n}+\ldots+T_{Kn} = n$ és a veszteség minimális legyen. Tegyük fel egy pillanatra, hogy az eloszlásokat – ismeretlen eltolások erejéig – ismerjük. Konkrétan ez azt jelenti, hogy nem ismerjük az eloszlások várható értékeit, de ismerjük az eloszlások szórásait és minden magasabb rendű momentumait. Ebben az esetben nincs értelme a $T_{1n}, \ldots, T_{Kn}$-t adatfüggően választani. A megfigyelések függetlensége miatt $L_{kn} = \sigma_k^2/T_{kn}$, ahol $\sigma_k^2 = \mathrm{Var}\left[Y_{k1}\right] = \mathrm{Var}\left[Y|X=k\right]$. Az egyszerűség kedvéért feltételezzük, hogy $\sigma_k^2 > 0$ minden k-ra. Ekkor nem nehéz látni, hogy $L_n = \max_k L_{kn}$-t az a $\{T_{kn}^*\}_{k=1}^K$ allokáció minimalizálja, amelyik az összes L_{kn} hibát (közelítőleg) egyenlővé teszi. Így (kerekítési kérdésektől eltekintve)

$$T_{kn}^* = n\frac{\sigma_k^2}{\Sigma^2} = n\lambda_k,$$

ahol $\Sigma^2 = \sum_{k'=1}^K \sigma_{k'}^2$ a szórásnégyzetek összege, míg $\lambda_k = \sigma_k^2/\Sigma^2$ a k. szórásnégyzet és a szórásnégyzetek összegének aránya. λ_k adja meg az *optimális allokációs arányt* a k. opcióhoz. Az ennek megfelelő veszteség

$$L_n^* = \frac{\Sigma^2}{n}.$$

Tehát az optimális allokáció kiszámolásához az eloszlásokról csak a szórásukat kell tudni. Az az eset, amikor minden szórás 0 (azaz $\Sigma^2 = 0$), érdektelen, így feltehető, hogy $\Sigma^2 > 0$.

1. gyakorlat. Gondoljuk meg, hogyan terjeszthető ki az optimális allokáció arra az esetre, amikor egyes (de nem minden) opciók szórása 0.

1.3.1. Megvalósítási lehetőségek

Egy jó szekvenciális $\mathcal{A}$ algoritmustól azt várjuk, hogy az L_n^* hibához közeli $L_n = L_n(\mathcal{A})$ hibát érjen el. Ezért a $L_n(\mathcal{A}) - L_n^*$ hibatöbbletet fogjuk tanulmányozni.

Mivel a k. opció hibája, L_{kn} csak csökkenhet attól, ha egy újabb megfigyelést kérünk a k. opcióról, az első egyszerű ötlet, hogy a következő megfigyelést arról a k. opcióról kérjük, amelyiknek a $\hat{\sigma}_{kn}^2/T_{kn}$ becsült hibája a legnagyobb az összes ilyen becsült hiba között. Itt $\hat{\sigma}_{kn}$ a k. opció szórásának egy becslése az eddigi minták alapján. Ezzel a módszerrel az a probléma, hogy a szórást alulbecsülhetjük, amely esetben az opciót sokáig nem fogjuk választani, ami megakadályozza, hogy a szórásbecslést pontosítsuk, végső soron nagy

hibatöbbletet eredményezve. Így egy a rabló-problémák felfedezés–kihasználás dilemmájához hasonló problémával állunk szemben. Erre egy egyszerű ellenszer biztosítani, hogy a becsült szórások konvergáljanak a valódiakhoz. Ez úgy tehető meg, ha az algoritmus kikényszeríti, hogy minden opciót korlátlan sokszor válasszunk az idő múlásával (amit a többkarú rabló irodalomban gyakran kényszerített mintavétel módszernek neveznek [76]). Ez többféle algoritmussal is megvalósítható:

UCB (Upper Confidence Bound) A többkarú rablóknál használthoz [8] hasonló konfidencia-intervallum felső határ típusú algoritmus [16].

GFSP (Greedy Forced Selections with Phases) Növekvő hosszúságú fázisokat vezetünk be. Minden fázis elején az algoritmus minden opciót pontosan egyszer választ, míg a fázis hátralévő részében minden opciót a fázis elején kiszámított, hozzá tartozó szórásnégyzet becslésével arányosan mintavételez. Ekkor a feladat megválasztani a megfelelő fázishosszúságokat, amelyek biztosítják, hogy a kényszerített választás aránya megfelelő rátával csökken a növekvő n horizont mellett. Ilyen algoritmust írtak le és elemeztek [41]-ben rétegzett mintavétellel összefüggésében. Míg a fázisok bevezetése lehetővé teszi a kényszerített választás arányának közvetlen szabályozását, az algoritmus nem inkrementális.

GAFS (Greedy Allocation with Forced Selections) Mohó allokáció kényszerített választással [7]. Inkrementális algoritmus, mely a legnagyobb becsült hibájú opciót választja, kivéve, ha valamelyik opció erősen alul-mintavételezett. Akkor egy ilyen opciót választ. Az opciók alul-mintavételezettségére használt definíció $T_{kn} \leq \alpha\sqrt{n}$ valamely $\alpha > 0$ állandóval. Lásd részletesen az 1.3.2. fejezetben.

1.3.2. GAFS algoritmus

A μ_k-nak, illetve σ_k^2-nak rendre a

$$\hat{\mu}_{kt} = \frac{1}{T_{kt}} \sum_{i=1}^{T_{kt}} Y_{ki}, \qquad \text{illetve} \qquad \hat{\sigma}_{kt}^2 = \frac{1}{T_{kt}} \sum_{i=1}^{T_{kt}} Y_{ki}^2 - \hat{\mu}_{kt}^2 \tag{1.6}$$

becslését használva a GAFS algoritmus formális leírása a következő:

GAFS algoritmus
Input: $\alpha > 0$ - konstans, ami szabályozza a felfedezés arányát
Inicializálás: Az első K lépésben válassz minden kart egyszer
Legyen $T_{kK} = 1$ és $\hat{\sigma}_{kK} = 0$ minden $1 \leq k \leq K$-ra
A $t = K+1,\ldots,n$. időpontokban:

Legyen $U_{t-1} = \arg\min_{1\leq k\leq K} T_{k,t-1}$

Legyen

$$X_t = \begin{cases} U_{t-1}, & \text{ha } T_{U_{t-1},t-1} < \alpha\sqrt{t-1}+1, \\ \arg\max_{1\leq k\leq K} \frac{\hat{\sigma}^2_{k,t-1}}{T_{k,t-1}} & \text{egyébként} \end{cases}$$

Válaszd az X_t kart és legyen $T_{kt} = T_{k,t-1} + \mathbf{I}_{\{X_t=k\}}$
Figyeld meg a $Y_t = Y_{X_t,T_{X_t,t}}$ kimenetet
Számítsd ki a $\hat{\mu}_{kt}$ becslést és $\hat{\sigma}^2_{kt}$-t (1.6) (csak $k = X_t$-re módosul)

3. Remark. *Az algoritmus egyetlen paramétere, α határozza meg a felfedezés minimális arányát. Általában $\alpha = 1$ megfelel. A szórásnégyzet becslések inkrementálisan is számíthatóak. Az* $\arg\min$- *és* $\arg\max$-*ban döntetlen esetén pl. mindig a szóba jövő legkisebb indexet választjuk.*

A GAFS algoritmus hibatöbbletére a következő eredmény ismert:

1.5. Theorem ([7]). *Alkalmas $C > 0$ konstansra $L_n(\mathcal{A}_{\mathrm{GAFS}}) - L^*_n \leq Cn^{-3/2}\sqrt{\log n}$.*

2. fejezet

Dimenzióredukció

2.1. Absztrakt

Adatelemzéseknél az alkalmazható eljárásokat jelentős mértékben befolyásolja a rendelkezésre álló adatok száma és dimenziója. A sokdimenziós adatok kezelése, ábrázolása komoly nehézségeket jelent, ezért fontos az adatok hatékonyabb, kisebb dimenziós ábrázolása. Sok esetben a sokdimenziós adatok valójában kisebb dimenziós térben is reprezentálhatóak lennének. Ehhez hasznos, ha az adatok struktúráját, az adatok fontos rejtett komponenseit megkíséreljük felderíteni, és ezáltal a sokdimenziós adatok kisebb dimenziós reprezentációját előállítani. Ez az összefoglaló – a teljesség igénye nélkül – a dimenzióredukció legfontosabb lineáris és nemlineáris eljárásairól ad egy áttekintést. A hangsúlyt a PCA-ra és ennek változataira helyezi, miközben említést tesz néhány további eljárásról is.

2.1.1. Kulcsszavak:

Dimenzióredukció, főkomponens-analízis, altér módszerek, kernel reprezentáció

2.2. Bevezetés

Adatelemzéseknél a megfelelő módszerek kiválasztását döntő módon befolyásolja, hogy milyen adatok állnak rendelkezésünkre, és az adatokról milyen ismeretünk van. A könnyen hozzáférhető és egyre növekvő számítási kapacitás, továbbá az olcsó adattárolási lehetőségek miatt egyre inkább érdemes a legkülönbözőbb területekről adatokat gyűjteni, minthogy ezen adatok a vizsgált területről – többnyire rejtve – fontos ismereteket tartalmaznak. Az adatelemzési eljárások feladata döntően éppen az, hogy segítse kinyerni ezeket a „rejtett" ismereteket, segítse a vizsgált témakörre vonatkozó új felismeréseket megfogalmazni.

Az adatgyűjtés könnyű és olcsó lehetősége következtében nagy adathalmazok keletkeztek, nagymennyiségű adat áll rendelkezésünkre, melyek elemzésére hatékony módszerek kidolgozása vált szükségessé. Minél teljesebb az adatok jellemzése, annál hatékonyabb eljárást tudunk kiválasztani.

Bár az adatok a legkülönbözőbb formában állhatnak rendelkezésünkre – lehetnek számszerű adataink, de lehetnek szöveges adataink, illetve képek formájában, stb. megjelenő adataink is – a továbbiakban adatokon mindig numerikus értékek együttesét fogjuk érteni. Az adathalmazunk minden elemét egy szám n-essel jellemezhetjük, vagyis egy olyan n-dimenziós $\mathbf{x}$ vektorral, mely vektor minden komponense egy (valós) szám. Abból indulunk ki tehát, hogy rendelkezésünkre áll $\{\mathbf{x}_i\}_{i=1}^N$ ahol $\mathbf{x}_i \in \mathrm{R}^n \ \forall i$-re. Az adatelemzési feladatok általában az adatok „struktúrájának", az adatokat leíró modellnek a felderítését jelentik, pl. az adatok „elhelyezkedését" keressük az n-dimenziós térben, vagy az egyes vektorok komponensei közötti esetleges kapcsolatok felderítése a cél.

Adataink legteljesebb jellemzését az jelentené, ha ismernénk az adatok eloszlását, ismernénk az adatok n-dimenziós sűrűség-, vagy eloszlásfüggvényét. Feltételezzük tehát, hogy adataink valószínűségi vektorváltozók konkrét realizációi, mely feltételezést leginkább az indokolja, hogy az adatok legtöbbször mérések eredményeként születnek, mely méréseket bizonytalanság is jellemez. Az adatok eloszlását általában nem ismerjük, legtöbbször mindössze az adathalmaz elemei állnak rendelkezésünkre, tehát az N db n-dimenziós vektor.

2.3. A dimenzió átka

Adott tehát egy n-dimenziós térben N mintapont. Fontos kérdés, hogy milyen viszonyban van egymással az adatok száma, N és az adatok dimenziója, n. Bár elvileg minden adatkomponensünk tetszőleges valós szám lehet, a valóságban az adatkomponesek csak diszkrét értékeket vehetnek fel, vagyis az n-dimenziós téren értelmezünk egy valamilyen finomságú térbeli rácsot. A lehetséges diszkrét értékek száma a rács felbontásától és a dimenziótól függ. Feltételezve, hogy minden dimenzió mentén M különböző értékünk lehet, egydimenziós esetben az összes lehetséges diszkrét adat száma M, kétdimenziós esetben M^2 és n-dimenziós esetben M^n. A lehetséges különböző adatok száma tehát a dimenzióval exponenciálisan nő. Ahhoz tehát, hogy n-dimenziós adatok mellett a teljes adatteret egyenletesen kitöltsük adatokkal, vagyis minden lehetséges diszkrét mintapont az adathalmazunkban legalább egyszer szerepeljen, a dimenzióval exponenciálisan növekvő számú adatra lenne szükség. Ha n még nem is tekinthető túl nagynak, legyen akár csak néhányszor 10, elfogadhatatlan számú mintapontra lenne szükségünk. Pl. M=10 és n=100 mellett ugyan a komponensenként lehetséges diszkrét értékek száma nem nagy, mégis nagyságrendileg 10^{100} mintapontra lenne szükségünk. Ezt szokás a dimenzió átkának nevezni ([Bel57]).

2.4. A dimenzióredukció alkalmazási területei

A probléma megoldását az adhatja, ha felismerjük, hogy adataink az n-dimenziós mintatér adott tartományában nem egyenletesen helyezkednek el. Bizonyos résztartományokban sűrűsödnek, míg más helyeken ritkán vagy egyáltalán nem fordulnak elő. Az is lehetséges, hogy az adataink valójában nem is n-dimenziósak, az n-dimenziós $\mathbf{x}$ vektoraink egy n-nél

sokkal kisebb, m-dimenziós altérben is reprezentálhatók lennének. A nehézséget csupán az okozza, hogy ezen m-dimenziós alteret meghatározó rejtett változókat nem ismerjük.

A dimenzióredukció egyik fő feladata az ilyen *rejtett változók meghatározása*, vagyis annak az *m-dimenziós altérnek* a meghatározása, melyben valójában megjelennek az adataink. A dimenzióredukció feladatát úgy is megfogalmazhatjuk, hogy az adatok olyan kisebb dimenziós reprezentációját keressük, mely a sokdimenziós térben adott adataink között meglévő valamilyen hasonlóságot, szomszédságot megtartja.

A rejtett m-dimenziós reprezentáció lehet az eredeti adataink pontos reprezentációja. Ugyanakkor számos esetben az m-dimenziós altérben nem tudjuk pontosan ábrázolni az adatokat, csak közelítőleg. Valójában ilyenkor az eredeti n-dimenziós térből egy olyan m-dimenziós altérbe történik az adatok vetítése, hogy az eredeti és a vetített reprezentációjú adatok eltérése valamilyen értelemben minimális legyen. Ebben a feladattípusban az adatok *közelítő, kisebb dimenziós reprezentációját* keressük olyan módon, hogy adott m mellett a lehető legkisebb hibájú közelítést kapjuk, vagy adott hibakorlát mellett a lehető legkisebb dimenziós alteret találjuk meg. Ez a feladat a *minimális hibájú adattömörítés.* Az adattömörítést általában az adatok hatékonyabb reprezentációja céljából végezzük, de alkalmazásával az is lehet a célunk, hogy az eredeti adatokból a későbbi feldolgozás szempontjából lényeges információt kiemeljük, és a lényegtelent elhagyjuk. Az adattömörítést ekkor *lényegkiemelés* érdekében végezzük.

Közelítő reprezentációt kapunk akkor is, amikor az adataink valójában pontosan ábrázolhatók lennének egy m-dimenziós altérben, azonban az adatokat terhelő zaj miatt hibamentsen ez mégsem tehető meg. Ilyenkor, ha megtaláljuk a legkisebb hibájú közelítő m-dimenziós reprezentációt, ez valójában az adatok zajmentes reprezentációjának felelhet meg. A feladat ekkor az adatokat terhelő *zaj csökkentése vagy eltüntetése.*

A dimenzióredukciót olyan céllal is alkalmazhatjuk, hogy az adatainkat könnyebben megjeleníthető formában ábrázoljuk. A sokdimenziós adatok könnyen áttekinthető megjelenítése, vizualizációja nehezen oldható meg, míg legfeljebb 3D-ig könnyen tudjuk szemléletesen ábrázolni az adatokat. Ilyen alkalmazásoknál a *lehető legkisebb, maximum 3-dimenziós közelítő reprezentációját keressük* az adatoknak, természetesen most is azon feltétel mellett, hogy a közelítés hibája adott hibadefiníció mellett a lehető legkisebb legyen.

A dimenzióredukció során az adatok olyan komponenseit keressük, olyan rejtett változókat keresünk, melyek valamilyen értelemben kitüntetettek, vagy önálló jelentéssel rendelkeznek. Ilyen értelemben az adatok dekomponálása is a dimenzióredukciós alapfeladathoz köthető. Az adatok *dekomponálása* mindig feltételez egy rejtett *adatmodellt*, mely modell megadja a komponenseket és az eredeti adatoknak a komponensekből való előállítási módját. Az adatok dekomponálásának egy speciális változata, ha komplex adatok statisztikailag független komponenseit szeretnénk meghatározni.

A dimenzióredukciós feladatok mindegyike értelmezhető úgy is, mint az adatok olyan reprezentációjának a keresése, amely során valamilyen mellékfeltételt is teljesíteni kell: keressük adott mellékfeltétel mellett az adatok optimális reprezentációját. A feladat tehát az

$$\mathbf{x}_i \in \mathrm{R}^n \rightarrow \mathbf{y}_i \in \mathrm{R}^m \quad m \leq n \tag{2.1}$$

leképezés megkeresése, úgy hogy valamilyen $C(\mathbf{x}_i,\mathbf{y}_i)$ kritérium (esetleg kritériumok) teljesüljön (teljesüljenek). A leképezés lehet lineáris, de általánosan akár nemlineáris is. Az ilyen típusú feladatoknál két nehézséggel találjuk magunkat szemben. Először is meg kell találnunk azt a kisebb dimenziós alteret, amelyben az eredeti adatok hatékonyan ábrázolhatók. Másodszor, ha a megfelelő alteret megtaláltuk, el kell végezzük a bemeneti adatoknak az altérre való vetítését.

Az előbbi az alteret meghatározó lineáris vagy nemlineáris transzformáció definiálását, az utóbbi a kiinduló adataink transzformálásának az elvégzését jelenti.

A dimenzióredukció tehát úgy is értelmezhető, hogy olyan vetítési irányt (irányokat) keresünk, mely irányokra történő vetítés adataink bizonyos fontos jellemzőit (pl. az adatok közötti hasonlóság, különbözőség, az adatokon értelmezett metrika, stb) megtartják, miközben a vetítési irányok által meghatározott altér dimenziója kisebb, mint az eredeti adatok dimenziója.

Ez az összefoglaló a dimenzióredukció néhány fontosabb eljárását tekinti át röviden. A lineáris eljárások között kitüntetett szerepe van a főkomponens-analízisnek (*principal component analysis, PCA*), mely többféle megközelítésből is származtatható. A nemlineáris eljárások tárháza igen széles. Itt csak néhány fontosabbat mutatunk be. Ezek között talán a legfontosabb a nemlineáris főkomponens-analízis (*NPCA*) és ennek is az ún. kernel változata, a kernel főkomponens-analízis (*KPCA*). A lineáris és nemlineáris főkomponens-analízissel rokon eljárások, a fő altér (*principal subspace*) meghatározó eljárások, melyek közül néhányat szintén bemutatunk röviden. A nemlineáris dimenzióredukciós eljárások közé tartoznak a (lineáris) főkomponens-analízishez hasonló fő görbe vagy fő felület (*principal curve*, *principal surface*) [Has89] eljárások, illetve a rejtett nemlineáris felületet közelítő lokális lineáris beágyazott felület meghatározását végző eljárás (*locally linear embedding LLE*) [Row00] is, melyekre a jelen összefoglaló nem tér ki, csak az irodalomra utal.

Az adatok komponensekre bontása, egyes komponensek megtartása, míg mások eldobása, szintén dimenzióredukcióként is értelmezhető. Az ilyen eljárások között nagy fontosságúak, sok gyakorlati alkalmazási feladatnál merülnek fel a független komponens analízis (*independent component analysis, ICA*) módszerei, melyeknek egész széles tárháza ismert [Hyv01]. Végül a dimenzióredukció témaköréhez is sorolhatók azok a módszerek is, ahol az adatok olyan vetítési irányát és így az eredetinél olyan kisebb dimenziós reprezentációját keresünk, hogy a hatékonyabb reprezentáció mellett még további célokat is megfogalmazunk. Ilyen további cél lehet pl., hogy az adatok egyes csoportjai a kisebb dimenziós térben jól elkülöníthetők legyenek. Ilyen eljárás az adatok kétosztályos osztályozását biztosító lineáris diszkrimináns analízis (*Linear Discriminant Analysis, LDA*), illetve ennek nemlineáris kiterjesztése [McL92]. Az összefoglaló – terjedelmi korlátok miatt – azonban ezekre az eljárásokra sem tér ki, csupán utal az irodalomra.

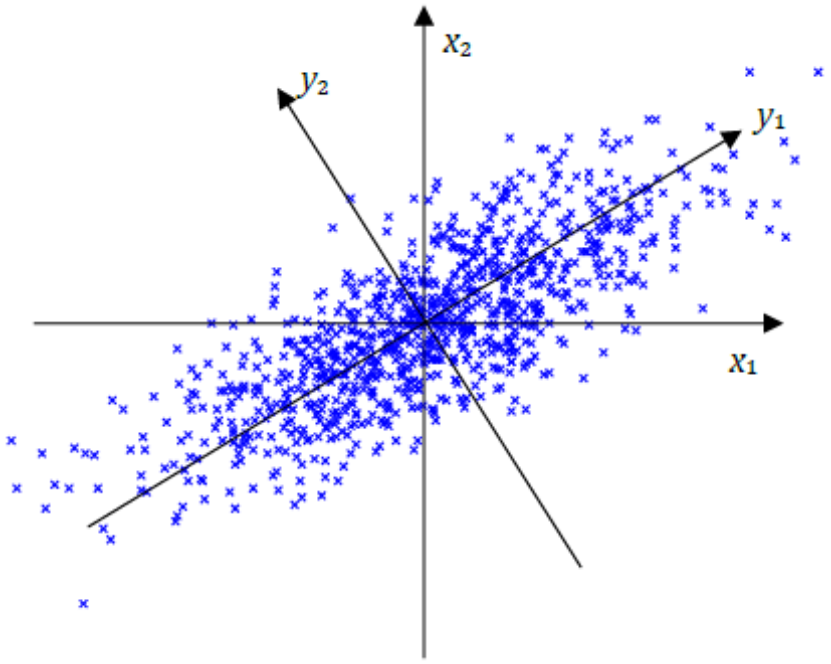
2.5. Főkomponens-analízis (PCA, KLT)

A hatékony ábrázolás az alkalmazási körtől függően különbözőképpen definiálható. Adattömörítésnél törekedhetünk arra, hogy az altérbe való vetítés során a vektor reprezentációnál a közelítő ábrázolásból adódó hiba minél kisebb legyen. Ebben a megközelítésben definiálni kell valamilyen hibakritériumot, pl. átlagos négyzetes hibát, majd egy olyan, az eredeti dimenziószámnál kisebb dimenziós altér megtalálása a feladat, amelybe vetítve a kiinduló vektort a kritérium szerint értelmezett reprezentációs hiba a lehető legkisebb lesz. Más feladatnál, pl. felismerési vagy osztályozási feladatot megelőző lényegkiemelésnél a reprezentáció akkor tekinthető hatékonynak, ha az altér dimenziója minél kisebb, miközben a közelítő reprezentációban mindazon információ megmarad, amely a felismeréshez, osztályozáshoz elegendő. Ebben az esetben tehát nem követelmény a kiinduló vektor minél kisebb hibájú reprezentálása, csupán arra van szükség, hogy olyan kisebb dimenziós ábrázolást kapjunk, amely a feladat szempontjából szükséges lényeges információkat megtartja.

E feladatok megoldására univerzális eljárás nem létezik. Általában a megfelelő altér feladat- és adatfüggő, tehát a tényleges feladattól függetlenül előre nem meghatározható, és mind az altér meghatározása, mind a transzformáció elvégzése meglehetősen számításigényes.

A lineáris adattömörítő eljárások között kitüntetett szerepe van a *Karhunen–Ločve-transzformáció*nak (KLT), amely az eredeti jeltér olyan ortogonális bázisrendszerét és az eredeti vektorok ezen bázisrendszer szerinti transzformáltját határozza meg, amelyben az egyes bázisvektorok fontossága különböző. Egy n-dimenziós térből kiindulva az új bázisvektorok közül kiválasztható a legfontosabb $m \leq n$ bázisvektor, amelyek egy m-dimenziós alteret határoznak meg. Egy vektornak ezen altérbe eső vetülete az eredeti vektorok közelítő reprezentációját jelenti, ahol a közelítés hibája átlagos négyzetes értelemben a legkisebb, vagyis a KLT a közelítő reprezentáció szempontjából a lineáris transzformációk között optimális bázisrendszert határoz meg.

A KL-transzformáció működését illusztrálja a 2.1. ábra. A transzformáció feladata az eredeti x_1, x_2 koordinátarendszerben ábrázolt adatokból kiindulva az y_1, y_2 koordinátarendszer megtalálása, majd az adatoknak ebben az új koordinátarendszerben való megadása. Látható, hogy míg az eredeti koordinátarendszerben a két komponens fontossága hasonló, addig az új koordinátarendszerben a két komponens szerepe jelentősen eltér: y_1 mentén jóval nagyobb tartományban szóródnak a mintapontok (a mintapontokról az y_1 irányú komponens sokkal többet mond el), mint y_2 mentén, tehát az egyes mintapontok közötti különbséget az y_1 koordináták jobban tükrözik. Amennyiben az adatok egydimenziós, közelítő reprezentációját kívánjuk előállítani célszerűen y_1-t kell meghagynunk és y_2-t eldobnunk; így lesz a közelítés hibája minimális. A KL-transzformáció szokásos elnevezése a matematikai statisztikában faktoranalízis vagy főkomponens-analízis (*principal component analysis, PCA*). Egy $\mathbf{x}$ vektor y_1 és y_2 irányú vetületeit főkomponenseknek is szokás nevezni. Közelítő reprezentációnál a főkomponensek közül csak a legfontosabbakat tartjuk meg, a többit eldobjuk. Az ábrán látható esetben ez azt jelenti, hogy egy $\mathbf{x}$ vektor legfontosabb főkomponense az y_1 tengely irányú vetülete. A főkomponens-analízis fontos



2.1. ábra. A Karhunen–Ločve-transzformáció

tulajdonsága, hogy az egyes komponenseket itt fontossági szempont szerint rangsorolva határozzuk meg.

A KL-transzformáció és optimalitása

A KL-transzformáció alapfeladata a következő: keressük meg azt az ortogonális (ortonormált) bázisrendszert, amely átlagos négyzetes értelemben optimális reprezentációt ad, majd e bázisrendszer segítségével végezzük el a transzformációt. Az eddigiekhez hasonlóan diszkrét reprezentációval dolgozunk, tehát a bemeneti jelet az n-dimenziós $\mathbf{x}$ vektorok képviselik, a transzformációt pedig egy $\mathbf{T}$ mátrixszal adhatjuk meg.

E szerint a transzformált jel ($\mathbf{y}$) előállítása:

$$\mathbf{y} = \mathbf{T}\mathbf{x}, \tag{2.2}$$

ahol a $\mathbf{T}$ transzformációs mátrix a $\boldsymbol{\varphi}_i$ bázisvektorokból épül fel:

$$\mathbf{T} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_n]^T \tag{2.3}$$

Mivel bázisrendszerünk ortonormált, ezért:

$$\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j = \delta_{ij}, \quad \text{ebből adódóan} \quad \mathbf{T}^T\mathbf{T} = \mathbf{I}, \quad \text{vagyis} \quad \mathbf{T}^T = \mathbf{T}^{-1}. \tag{2.4}$$

Feladatunk legyen a következő: $\mathbf{x}$ közelítő reprezentációját ($\hat{\mathbf{x}}$) akarjuk előállítani úgy, hogy a közelítés négyzetes hibájának várható értéke minimális legyen. Mivel $\mathbf{x}$ előállítható, mint

a $\boldsymbol{\varphi}_i$ bázisvektorok lineáris kombinációja

$$\mathbf{x} = \sum_{i=1}^{n} y_i \boldsymbol{\varphi}_i \,, \tag{2.5}$$

ahol y_i a $\boldsymbol{\varphi}_i$ irányú komponens nagysága, és mivel a közelítő reprezentáció

$$\hat{\mathbf{x}} = \sum_{i=1}^{n} y_i \boldsymbol{\varphi}_i \qquad m \leq n, \tag{2.6}$$

a négyzetes hiba várható értéke felírható az alábbi formában:

$$\varepsilon^2 = E\left\{\|\mathbf{x} - \hat{\mathbf{x}}\|_F^2\right\} = E\left\{\left\|\sum_{i=1}^{n} y_i \boldsymbol{\varphi}_i - m\sum_{i=1}^{m} y_i \boldsymbol{\varphi}_i\right\|_F^2\right\} = \sum_{i=m+1}^{n} E\left\{(y_i)\right\}. \tag{2.7}$$

ahol $\|.\|_F$ a Frobenius normát jelöli. Továbbá, mivel

$$y_i = \boldsymbol{\varphi}_i^T \mathbf{x}, \tag{2.8}$$

a következő összefüggés is a fenti hibát adja meg:

$$\varepsilon^2 \sum_{i=m+1}^{n} E\left\{(\boldsymbol{\varphi}_i^T \mathbf{x})(\mathbf{x}^T \boldsymbol{\varphi}_i)\right\} = \sum_{i=m+1}^{n} \boldsymbol{\varphi}_i^T E\left\{\mathbf{x}\mathbf{x}^T\right\} \boldsymbol{\varphi}_i = \sum_{i+m+1}^{n} \boldsymbol{\varphi}_i^T \mathbf{R}_{\mathbf{xx}} \boldsymbol{\varphi}_i, \tag{2.9}$$

ahol $\mathbf{R}_{\mathbf{xx}}$ az $\mathbf{x}$ bemenet autokorrelációs mátrixa. A továbbiakban feltételezzük, hogy $E\{\mathbf{x}\}$=$\mathbf{0}$, ekkor $\mathbf{R}_{\mathbf{xx}}$ helyett $\mathbf{C}_{\mathbf{xx}}$, vagyis $\mathbf{x}$ kovarianciamátrixa szerepel a négyzetes hiba kifejezésében.

Ezek után keressük azt a $\boldsymbol{\varphi}_i$ bázist, amely mellett ε^2 minimális lesz. Ehhez az szükséges, hogy teljesüljön a

$$\mathbf{C}_{\mathbf{xx}} \boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i, \tag{2.10}$$

összefüggés [Dia96], vagyis a KLT bázisrendszerét alkotó $\boldsymbol{\varphi}_i$ vektorok a bemeneti jel autokovariancia mátrixának sajátvektorai legyenek. A közelítő, m-dimenziós reprezentáció esetén elkövetett hiba ilyenkor

$$\varepsilon^2 = \sum_{i=m+1}^{n} \boldsymbol{\varphi}_i^T \mathbf{C}_{\mathbf{xx}} \boldsymbol{\varphi}_i = \sum_{i=m+1}^{n} \boldsymbol{\varphi}_i^T \lambda_i \boldsymbol{\varphi}_i = \sum_{i=m+1}^{n} \lambda_i, \tag{2.11}$$

ahol a λ_i értékek az autokovariancia mátrix sajátértékei, ahol $\lambda_i \geq 0$ $\forall i$-re. Minimális hibát nyilvánvalóan akkor fogunk elkövetni, ha a (2.11) összefüggésben a λ_i sajátértékek (i=m+1,...,n) a mátrix legkisebb sajátértékei, vagyis a közelítő, m-dimenziós reprezentációnál az autokovariancia mátrix első m legnagyobb sajátértékéhez tartozó sajátvektort, mint m-dimenziós bázist használjuk fel. A bemeneti jel ezen vektorok irányába eső vetületei lesznek a főkomponensek (innen ered a főkomponens-analízis elnevezés). Megjegyezzük,

hogy a KL-transzformáció korrelálatlan komponenseket eredményez, vagyis a transzformált jel autokovariancia mátrixa diagonál mátrix, melynek főátlójában a λ_i sajátértékek vannak.

A KLT egy kétlépéses eljárás: először a bemeneti jel autokovariancia mátrixát, és ennek sajátvektorait és sajátértékeit kell meghatározni, majd ki kell választani a legnagyobb m sajátértéknek megfelelő sajátvektort, amelyek a megfelelő altér bázisvektorait képezik. A bázisrendszer ismeretében lehet elvégezni második lépésként az adatok transzformációját.

A sajátvektor(ok) meghatározására számos módszer áll rendelkezésünkre. A klasszikus megoldások a $\mathbf{C}$ kovariancia mátrixból indulnak ki, míg más eljárások közvetlenül az adatokból dolgoznak, $\mathbf{C}$ meghatározása nélkül.

2.5.1. Főkomponens- és altér-meghatározó eljárások

A legfontosabb főkomponens irányát meghatározó sajátvektor a fentiektől eltérően, szélsőérték-kereső eljárás eredményeként is származtatható, minthogy a bemenő sztochasztikus vektorfolyamat mintáinak a legnagyobb sajátvektor irányában vett vetülete várható értékben maximumot kell adjon. Keressük tehát $f(\mathbf{w}) = E\{y^2\}$ maximumát $\mathbf{w}$ függvényében azzal a feltételezéssel, hogy $\|\mathbf{w}\|_F = 1$.

$$f(\mathbf{w}) = \frac{E\{y^2\}}{\mathbf{w}^t\mathbf{w}} = \frac{\mathbf{w}^t\mathbf{C}\mathbf{w}}{\mathbf{w}^t\mathbf{w}} \tag{2.12}$$

A (2.12) összefüggést Rayleigh-hányadosnak is nevezik [Gol96b], mely ha $\mathbf{w}$ a $\mathbf{C}$ mátrix egy sajátvektora, a hányados a megfelelő sajátértéket adja. Ebből is következik, hogy (2.12) $\mathbf{w}$ szerinti maximuma a legnagyobb, minimuma a legkisebb sajátértéket eredményezi.

A mátrix explicit ismerete nélkül is van mód a főkomponensek meghatározására. Ebben az esetben olyan iteratív eljárást alkalmazhatunk, mely közvetlenül az adatokból, a kovariancia mátrix kiszámítása nélkül adja meg a legnagyobb sajátértékhez tartozó sajátvektort és legfontosabb főkomponenst. Az iteratív eljárás az ún. Oja-algoritmus [Oja82]:

$$\mathbf{w}(k+1) = [\mathbf{w}(k) + \mu\mathbf{x}(k)y(k)]\left[1 - \mu y^2(k) + O(\mu^2)\right] \approxeq \mathbf{w}(k) + \mu y(k)\left[\mathbf{x}(k) - \mathbf{w}(k)y(k)\right], \tag{2.13}$$

ahol $\mathbf{w}(k)$ az iteratív algoritmus eredménye a k-adik iterációban, μ pedig egy skalár együttható. Az eljárás megfelelő μ megválasztása esetén bizonyítottan konvergál a legfontosabb sajátvektorhoz, bár a konvergencia általában elég lassú.

A Rayleigh-hányados maximalizálása, illetve az Oja-algoritmus csak a legfontosabb sajátvektort eredményezi. A különböző alkalmazásokban a legfontosabb sajátvektornak és az ebbe az irányba eső főkomponensnek a meghatározása általában nem elegendő. Olyan hálózatot szeretnénk kapni, amely n-dimenziós bemenetből kiindulva az m legfontosabb sajátvektor ($m \leq n$) meghatározására képes.

A (2.13) összefüggés alapján olyan hierarchikus számítási rendszer alakítható ki, mely egymást követően számítja ki rendre a legfontosabb, a második legfontosabb, stb. sajátvektorokat. Ez a megközelítés mind a Rayleigh-hányados alapján, mind az Oja-algoritmussal

végzett számításnál alkalmazható. Az Oja-algoritmusból kiindulva a hierarchikus számítási eljárás egyetlen iteratív összefüggéssel is leírható:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu \left[\mathbf{y}(k)\mathbf{x}^T(k) - \mathrm{LT}(\mathbf{y}(k)\mathbf{y}^T(k))\mathbf{W}(k) \right], \tag{2.14}$$

ahol $\mathrm{LT}(\mathbf{A})$ az $\mathbf{A}$ mátrixból képezett alsó háromszögmátrixot jelöli. Ez az ún. általánosított Hebb-algoritmus (Generalized Hebbian Algorithm, GHA) [San89]. Az összefüggésben $\mathbf{W}$ az autokovariancia mátrix összes sajátvektorát tartalmazó mátrix, legalábbis megfelelő μ megválasztása esetén az eljárás ehhez a mátrixhoz konvergál.

A főkomponenseket meghatározó eljárások mellett sokszor elegendő, ha nem a tényleges főkomponenseket, vagyis a legfontosabb sajátvektorok irányába eső vetületeket határozzuk meg, hanem csupán azt az alteret és ebbe az altérbe eső vetületet, amelyet az első m legfontosabb sajátvektor feszít ki. Az alteret nemcsak a sajátvektorok határozzák meg, hanem bármely bázisa. Azokat az eljárásokat, amelyek az alteret és a bemeneti vektorok altérbe eső vetületeit meghatározzák, de a sajátvektorokat nem, altér (*subspace*) eljárásoknak nevezzük.

A (2.14) összefüggéssel megadott Oja-algoritmus egyszerűen módosítható olyan módon, hogy ne a legfontosabb főkomponens meghatározását végezze, hanem az első m sajátvektor által kifeszített altérbe vetítsen. Az algoritmus tehát átlagos négyzetes értelemben minimális hibájú közelítést eredményez, ha az eredeti Oja-szabályt egy m-dimenziós $\mathbf{y}$ kimeneti vektorra alkalmazzuk. Az eredmény az Oja-általánosított szabály [Oja82]:

$$\mathbf{\Delta W} = \mu \left[\mathbf{W}\mathbf{x}\mathbf{x}^T - \left(\mathbf{W}\mathbf{x}\mathbf{x}^T\mathbf{W}^T \right) \mathbf{W} \right], \tag{2.15}$$

ahol $\mathbf{W} = \left[\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M \right]^T$ az m-kimenetű rendszer súlyvektoraiból, mint sorvektorokból képezett mátrix.

Az Oja-altér hálózat egy n-bemenetű−m-kimenetű hálózat. Mivel az Oja-altér háló súlyvektorai nem a sajátvektorokhoz konvergálnak, hanem a sajátvektorok által kifeszített tér egy bázisához, az általánosított Oja-szabályt Oja-altér szabálynak is szokás nevezni.

Az altér feladatot az előbbiektől lényegesen eltérő szemlélet alapján is meg tudjuk oldani. Minthogy az altérre vetítés egy lineáris transzformáció, az m-dimenziós altérből az eredeti n-dimenziós térbe való „visszavetítés" is elvégezhető egy lineáris transzformációval. A két transzformáció eredőjeként a kiinduló adatoknak az eredeti n-dimenziós térbeli közelítő reprezentációját kapjuk. A vetítés eredménye:

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \tag{2.16}$$

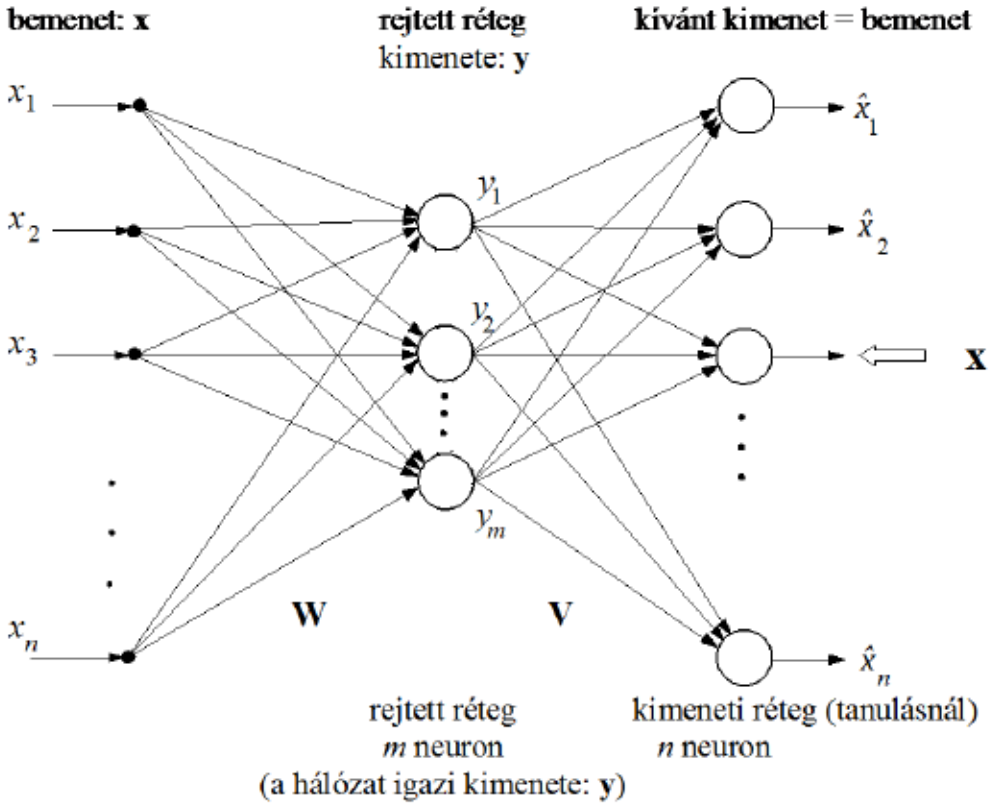
ahol $\mathbf{W}$ a vetítés $m \times n$ dimenziós mátrixa, míg a visszavetítésé:

$$\hat{\mathbf{x}} = \mathbf{V}\mathbf{y}, \tag{2.17}$$

ahol a visszavetítést a $\mathbf{V}$ $n \times m$ dimenziós mátrix definiálja. Általában $\mathbf{VW} \neq \mathbf{I}$, vagyis $\hat{\mathbf{x}} \neq \mathbf{x}$. Ekkor olyan $\mathbf{W}$ és $\mathbf{V}$ meghatározása a feladat, hogy hasonlóan a (2.7) összefüggésben megfogalmazott esethez, a rekonstrukció hibája

$$\varepsilon^2 = E\left\{ \|\mathbf{x} - \hat{\mathbf{x}}\|_F^2 \right\} \tag{2.18}$$

legyen minimális. Ez egy olyan többrétegű perceptron neurális hálózattal [Hor06] is megvalósítható, mely neurális hálózat lineáris neuronokból épül fel (hiszen $\mathbf{x}$ és $\mathbf{y}$, illetve $\mathbf{y}$ és $\hat{\mathbf{x}}$ között lineáris transzformációt kell megvalósítani) és autoasszociatív módon működik, vagyis adott bemenetre válaszként magát a bemenetet várjuk. Amennyiben a rejtett rétegbeli neuronok száma (m) kisebb, mint a bemenetek (és ennek megfelelően a kimenetek) száma (n), akkor a rejtett rétegbeli neuronok kimenő értékei a bemenet tömörített (közelítő) reprezentációját adják (ld. 2.2. ábra).



2.2. ábra. Lineáris többrétegű perceptron, mint adattömörítő autoasszociatív háló

A rejtett réteg képezi a háló „szűk keresztmetszetét". Ha a hálót a szokásos hibavisszaterjesztéses algoritmussal tanítjuk, a háló által előállított kimenet ($\mathbf{y}$) átlagos négyzetes értelemben közelíti a háló bemeneti jelét ($\mathbf{x}$). A háló kimeneti rétege a rejtett rétegbeli m-dimenziós reprezentációból állítja vissza az n-dimenziós kimenetet, tehát a rejtett réteg kimenetén a bemenőjel kisebb dimenziós altérbe vett vetületét kapjuk meg, olyan módon, hogy e közelítő ábrázolásból az eredeti jel a legkisebb átlagos négyzetes hibával állítható vissza. Az altér lineáris neuronok mellett bizonyítottan [Bal89] a megfelelő KLT alteret jelenti, de az altérben a bázisvektorok – a $\mathbf{W}$ transzformációs mátrix sorvektorai – nem feltétlenül lesznek a sajátvektorok. Megjegyezzük, hogy ez az autoasszociatív neuronhálós megoldás is alkalmas lehet a valódi főkomponensek meghatározására, amennyiben a háló csak egy rejtett neuront tartalmaz. Ebben az esetben a tanítás során ennek a neuron-

nak a kimenete a legfontosabb főkomponenshez, a neuron súlyvektora pedig a legnagyobb sajátértékhez tartozó sajátvektorhoz fog konvergálni. Ha több főkomponenst szeretnénk meghatározni, akkor az előbbiekben említett hierarchikus rendszerrel itt is meghatározhatók az egymást követő sajátvektorok és főkomponensek. Ehhez mindössze arra van szükség, hogy egyetlen háló helyett m hálót alkalmazzunk, melyek bemeneti vektorai az eredeti bemenetkből a már meghatározott főkomponensek figyelembevételével származtatott bemeneteket kapják.

2.6. Nemlineáris dimenzióredukciós eljárások

A PCA lineáris transzformációt végez, a koordinátarendszer olyan forgatását végzit, melynek eredményeként az eredeti koordinátarendszerről a $\mathbf{C}$ mátrix sajátvektorainak koordinátarendszerére térünk át. Az új koordinátarendszerben az egyes irányok „fontosságát" a sajátvektorokhoz tartozó sajátértékek adják meg. Ha a sajátértékek jelentősen különböznek, lehetőségünk van az adatok dimenzióját redukálni úgy, hogy átlagos négyzetes eltérés értelemben a redukció következtében elkövetett hiba minimális legyen. A legnagyobb m sajátértékhez tartozó, „legfontosabb" sajátvektort tartjuk meg, és az így kapott m-dimenziós térre vetítjük a mintapontjainkat. Az altér algoritmusok ugyan nem a sajátvektorokat határozák meg, de itt is a sajátvektorok által meghatározott altér megtalálása a feladat, melyet szintén egy megfelelő lineáris transzformáció biztosít.

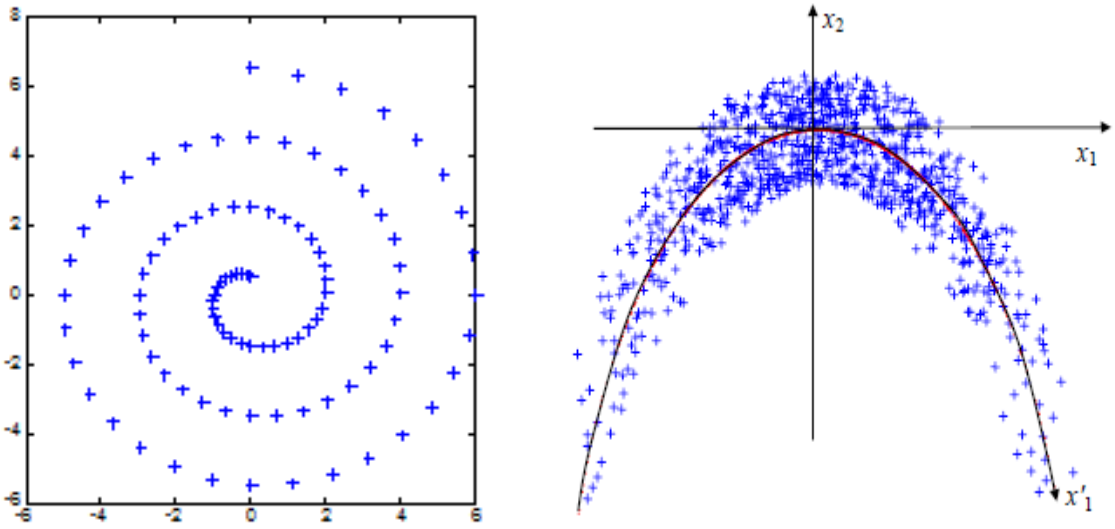
Ha a sajátértékek közel azonosak, akkor a sajátvektorok meghatározása egyre kevésbé lesz definit, ráadásul nem találunk olyan kitüntetett, „fontos" irányokat, melyek szerepe a többi iránynál nagyobb az adatok reprezentálása során. Szemléletes kétdimenziós példa erre az estre, ha az adatok egy spirál vagy egy parabola mentén helyezkednek el (2.3. ábra).

Az ábrából látható, hogy az adatok valójában egydimenziósak vagy közel egydimenziósak, a rejtett dimenzió meghatározása azonban lineáris transzformációval nem lehetséges. Hasonló helyzetet mutat a 2.4. ábra, ahol a háromdimenziós ún. *swissroll* adatokat ábrázoltuk. Az adatok valójában itt sem háromdimenziósak, egy rejtett kétdimenziós altér (nemlineáris felület) megtározása, és erre az altérre való vetítés ad lehetőséget a dimenzióredukcióra.

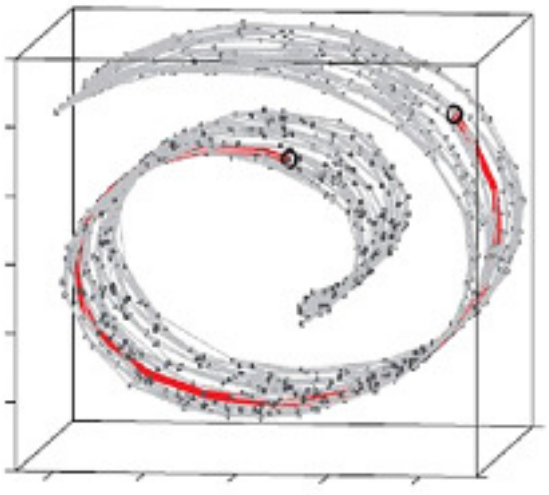
Az ilyen feladatoknál az adatok az eredeti n-dimenziós térbeni reprezentációnál egy kisebb m-dimenziós térben is reprezentálhatók, akár hibamentesen is, amenynyiben a rejtett m-dimenziós teret meg tudjuk határozni. A megoldáshoz az adatoknak olyan nemlineáris transzformációjára van szükség, hogy a transzformált térben az adattömörítés, dimenzióredukció már lineáris eljárásokkal elvégezhető legyen. A $\mathbf{\Phi}$ nemlineáris transzformáció az eredeti bemeneti térből egy ún. jellemzőtérbe transzformál:

$$\mathbf{\Phi} : \mathrm{R}^n \to F, \qquad \mathbf{x} \mapsto \mathbf{X} = \mathbf{\Phi}(\mathbf{x}) \tag{2.19}$$

A jellemzőtérben – ha a nemlineáris transzformációt megfelelően választottuk meg – már alkalmazhatók a lineáris módszerek, pl. a főkomponens-analízis (PCA). A teljes eljárás



2.3. ábra. Kétdimenziós mintakészletek, ahol a lineáris dimenzióredukció nem működik



2.4. ábra. Nemlineáris dimenzióredukció

azonban az eredeti térben nemlineáris, ezért szokás nemlineáris főkomponens-analízisnek (NPCA) is nevezni.

A nemlineáris főkomponens-analízis eljárásoknál is – hasonlóan a lineáris eljárásokhoz – olyan új koordinátarendszert keresünk, melynek egyes koordinátái jelentős mértékben eltérő fontosságúak az adatok előállításában. A kétféle eljárás közötti alapvető különbség, hogy itt a megfelelő transzformáció keresését nem korlátozzuk a lineáris transzformációk körére. Ezt elvben úgy tesszük, hogy a nemlineáris feladat megoldását két lépésben végezzük el: először az adatokat a jellemzőtérre képezzük le alkalmasan megválasztott nemlineáris transzformációval, majd a lineáris PCA-t a jellemzőtérben alkalmazzuk.

A módszer nehézségét már lineáris eljárásnál is az okozta, hogy a transzformáció bázisa a kiinduló adatok függvénye. Nemlineáris esetben a megfelelő transzformáció megtalálása és hatékony megvalósítása még nehezebb feladat.

A következőkben egy nemlineáris eljárást mutatunk be. Az itt alkalmazott nemlineáris transzformáció általában a bemeneti térnél sokkal nagyobb dimenziós jellemzőteret eredményez, azonban a jellemzőtérbeli főkomponens-analízist nem ebben a térben, hanem az ebből származtatott ún. kernel térben tudjuk megoldani. Itt tehát nincs szükség a jellemzőtérbeli transzformáció explicit definiálására és a jellemzőtérbeli reprezentáció meghatározására. Az ún. kernel trükk segítségével ugyanis a jellemzőtérbeli főkomponens-analízis elvégezhető a kernel térben is. Az ún. kernel PCA célja az adatokban meglévő rejtett (nemlineáris) struktúra meghatározása. A kernel PCA tehát nem feltétlenül dimenzióredukcióra szolgál, bár a nemlineáris dimenzióredukciónak is hatékony eszköze.

2.6.1. Kernel PCA

A PCA során a bemeneti térben keresünk főkomponenseket úgy, hogy a bemenetek megfelelő lineáris transzformációját végezzük. A kernel PCA ezzel szemben nem a bemeneti térben keres főkomponenseket, hanem előbb a bemeneti vektorokat nemlineáris transzformációval egy ún. jellemzőtérbe transzformálja, és itt keres főkomponenseket.

Az eljárás bemutatásához a következő jelölésekből induljunk ki. Jelöljük a bemeneti térből a jellemzőtérbe való nemlineáris transzformációt Φ-vel. A bemeneti tér lehet pl. a valós szám n-esek tere, R^n, ekkor a nemlineáris transzformáció R^n-ből egy F jellemzőtérbe képez le:

$$\mathbf{\Phi} : \mathrm{R}^n \to F, \qquad \mathbf{x} \mapsto \mathbf{X} = \mathbf{\Phi}(\mathbf{x}) \tag{2.20}$$

Az F jellemzőtér tetszőlegesen sokdimenziós, akár végtelen dimenziós tér is lehet. Tételezzük fel, hogy az F térben is fennáll, hogy $\sum_{k=1}^{N} \mathbf{\Phi}\left(\mathbf{x}_k\right) = \mathbf{0}$, ahol N a bemeneti vektorok száma. Becsüljük a jellemzőtérbeli kovarianciamátrixot a véges számú mintapont (jellemzőtérbeli vektor) alapján:

$$\bar{\mathbf{C}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{\Phi}\left(\mathbf{x}_j\right) \mathbf{\Phi}\left(\mathbf{x}_j\right)^T . \tag{2.21}$$

A jellemzőtérbeli főkomponensek meghatározásához először most is meg kell határoznunk a kovarianciamátrix nemnulla sajátértékeit és a megfelelő sajátvektorokat, melyek kielégítik

a szokásos sajátvektor-sajátérték egyenletet:

$$\lambda \mathbf{V} = \bar{\mathbf{C}}\mathbf{V}, \tag{2.22}$$

majd a jellemzőtérbeli főkomponenseket a $\mathbf{\Phi}(\mathbf{x})$ jellemzőtérbeli vektorok és az egységnyi hosszúságúra normált $\mathbf{V}$ sajátvektorok skalár szorzataként kapjuk.

A sajátértékek és a sajátvektorok meghatározásához hasznos, ha felhasználjuk, hogy a $\bar{\mathbf{C}}$ kovarianciamátrix sajátvektorai a jellemzőtérbeli vektorok által kifeszített térben vannak:

$$\mathbf{V} = \sum_{i=1}^{N} \alpha_i \mathbf{\Phi}\left(\mathbf{x}_i\right), \tag{2.23}$$

tehát léteznek olyan α_i (i=1,...,N) együtthatók, melynek segítségével a sajátvektorok előállíthatók a bemeneteket a jellemzőtérben reprezentáló vektorok súlyozott összegeként. A (2.23) összefüggés felhasználásával azonban meg tudjuk mutatni, hogy a jellemzőtérbeli főkomponensek anélkül is meghatározhatók, hogy a bemeneti vektorok jellemzőtérbeli reprezentációját meghatároznánk.

Ennek érdekében tekintsük a következő egyenletet:

$$\lambda \mathbf{\Phi}^T\left(\mathbf{x}_k\right)\mathbf{V} = \mathbf{\Phi}^T\left(\mathbf{x}_k\right)\bar{\mathbf{C}}\mathbf{V}, \quad k = 1, \ldots, N. \tag{2.24}$$

Helyettesítsük ebbe az egyenletbe $\bar{\mathbf{C}}$ (2.17) és $\mathbf{V}$ (2.23) összefüggését. Ekkor minden k=1,..., N-re a következőt kapjuk:

$$\lambda \sum_{i=1}^{N} \alpha_i \mathbf{\Phi}^T\left(\mathbf{x}_k\right)\mathbf{\Phi}\left(\mathbf{x}_i\right) = \frac{1}{N}\sum_{i=1}^{N} \alpha_i \mathbf{\Phi}^T\left(\mathbf{x}_k\right)\sum_{j=1}^{N}\mathbf{\Phi}\left(\mathbf{x}_j\right)\mathbf{\Phi}^T\left(\mathbf{x}_j\right)\mathbf{\Phi}\left(\mathbf{x}_i\right). \tag{2.25}$$

Vegyük észre, hogy ebben az összefüggésben a jellemzőtérbeli $\mathbf{\Phi}(\mathbf{x})$ vektorok mindig csak skalár szorzat formájában szerepelnek.

Definiáljunk egy $N \times N$méretű $\mathbf{K}$ kernel mátrixot, melynek (i,j)-edik eleme:

$$K_{ij} = K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{\Phi}^T\left(\mathbf{x}_i\right)\mathbf{\Phi}\left(\mathbf{x}_j\right). \tag{2.26}$$

Ezzel a (2.25) összefüggés az alábbi tömör formában is felírható:

$$N\lambda \mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^2\boldsymbol{\alpha}, \tag{2.27}$$

ahol az $\boldsymbol{\alpha}$ oszlopvektor az α_i i=1,...,N együtthatókból áll. $\mathbf{K}$ szimmetrikus mátrix, és ha megoldjuk a következő sajátvektor-sajátérték problémát:

$$N\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{2.28}$$

ahol az $\boldsymbol{\alpha}$ vektorok $\mathbf{K}$ sajátvektorai és a $N\lambda$ értékek a sajátértékek, a megoldás kielégíti a (2.27) egyenletet is. Jelöljük $\mathbf{K}$ nemnulla sajátértékeit nagyság szerint sorbarendezve $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$-vel, a hozzájuk tartozó sajátvektorokat pedig $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(N)}$-vel, és legyen λ_r az első (legkisebb) nemnulla sajátérték. (Ha feltételezzük, hogy $\mathbf{\Phi}(\mathbf{x})$ nem

azonosan $\mathbf{0}$, akkor mindig léteznie kell egy ilyen λ_r-nek.) Normalizáljuk az $\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(N)}$ sajátvektorokat, hogy az F térben a következő egyenlőség teljesüljön $k{=}r,\ldots,N$-re:

$$\mathbf{V}^{(k)^T}\mathbf{V}^{(k)} = 1. \tag{2.29}$$

Ez a következő normalizálási feltételt szabja az $\boldsymbol{\alpha}$ sajátvektorokra:

$$\begin{aligned} 1 &= \sum_{i,j=1}^{N} \alpha_i^{(k)} \alpha_j^{(k)} \boldsymbol{\Phi}^T(\mathbf{x}_i)\, \boldsymbol{\Phi}(\mathbf{x}_j) \\ &= \sum_{i,j=1}^{N} \alpha_i^{(k)} \alpha_j^{(k)} K_{ij} = \boldsymbol{\alpha}^{(k)^T} \mathbf{K} \boldsymbol{\alpha}^{(k)} \\ &= \lambda_k \boldsymbol{\alpha}^{(k)^T} \boldsymbol{\alpha}^{(k)}. \end{aligned} \tag{2.30}$$

A főkomponensek meghatározása után szükségünk van még a jellemzőtérbeli vektorok sajátvektorok szerinti vetítésére. Legyen $\mathbf{x}$ egy tesztpont, $\boldsymbol{\Phi}(\mathbf{x})$ képpel F-ben, ekkor

$$\mathbf{V}^{(k)^T}\boldsymbol{\Phi}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^{(k)} \boldsymbol{\Phi}^T(\mathbf{x}_i)\, \boldsymbol{\Phi}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^{(k)} K(\mathbf{x}_i, \mathbf{x}). \tag{2.31}$$

A jellemzőtérbeli főkomponens tehát a közvetlenül a kernel értékek függvényében kifejezhető, anélkül, hogy a $\boldsymbol{\Phi}(\mathbf{x})$ nemlineáris leképezéseket meg kéne határozni. Tehát itt is a kernel trükköt alkalmazhatjuk, ha a nemlineáris PCA számítását nem a $\boldsymbol{\Phi}(\mathbf{x})$ nemlineáris leképezések rögzítésével, hanem a $\mathbf{K}$ mátrix (a kernel függvény) megválasztásával végezzük. A kernel PCA-nál tehát nem a $\boldsymbol{\Phi}(\mathbf{x})$ nemlineáris leképezésekből, hanem a kernel függvényből indulunk ki. A kernel függvény implicit módon definiálja a jellemzőtérbeli leképezést.

Összefoglalva a következő teendőink vannak a főkomponensek meghatározása során. Először meg kell választanunk a kernel függvényt, majd meg kell határoznunk a $\mathbf{K}$ mátrixot. Ennek a mátrixnak kell kiszámítanunk az $\boldsymbol{\alpha}^{(k)}$ sajátvektorait. A sajátvektorok normalizálását követően határozhatjuk meg a bemeneti vektorok jellemzőtérbeli főkomponenseit a (2.31) összefüggés felhasználásával.

Az eljárás fő előnye abban rejlik, hogy a $\boldsymbol{\Phi}(\mathbf{x})$ függvény ismeretére nincs szükségünk, továbbá, hogy míg az eredeti PCA során a kovarianciamátrix mérete a bemeneti dimenziótól függ, addig itt a $\mathbf{K}$ mátrix méretét a tanítópontok száma határozza meg. Lineáris PCA-nál legfeljebb n sajátvektort és így n főkomponenst találunk, ahol n a bemeneti vektorok dimenziója. Kernel PCA-nál maximum N nemnulla sajátértéket kaphatunk, ahol N a mintapontok száma.

A nulla várhatóérték biztosítása a jellemzőtérben A korábbiakban tényeként kezeltük, hogy az F térben igaz a $\sum_{k=1}^{N} \boldsymbol{\Phi}(\mathbf{x}_k) = \mathbf{0}$ megállapítás. Ez nyilvánvalóan nem lehet igaz minden $\boldsymbol{\Phi}(\mathbf{x})$ függvényre, így szükségünk van arra, hogy a $\boldsymbol{\Phi}(\mathbf{x})$ jellemzőtérbeli

vektorokat is **0** átlagértékűvé transzformáljuk. Ez megoldható, ha a vektorokból kivonjuk az átlagukat:

$$\tilde{\mathbf{\Phi}}(\mathbf{x}_i) = \mathbf{\Phi}(\mathbf{x}_i) - \frac{1}{N}\sum_{k=1}^{N}\mathbf{\Phi}(\mathbf{x}_k). \tag{2.32}$$

Az eddigi megállapítások szerint most ez alapján kell meghatározni a kovarianciamátrixot, illetve a

$$\tilde{K}_{ij} = \tilde{\mathbf{\Phi}}^T(\mathbf{x}_i) \cdot \tilde{\mathbf{\Phi}}(\mathbf{x}_j) \tag{2.33}$$

mátrixot az F térben. Az így kapott $\tilde{\mathbf{K}}$ mátrix sajátérték-sajátvektor rendszerét kell meghatároznunk:

$$\tilde{\lambda}\tilde{\boldsymbol{\alpha}} = \tilde{K}\tilde{\boldsymbol{\alpha}}, \tag{2.34}$$

ahol $\tilde{\alpha}$ a sajátvektorok együtthatóit tartalmazza a következő formában:

$$\tilde{\mathbf{V}} = \sum_{i=1}^{N}\tilde{\alpha}\tilde{\mathbf{\Phi}}(\mathbf{x}_i). \tag{2.35}$$

A $\tilde{\mathbf{K}}$ mátrix kiszámítása a definíciós összefüggés szerint azonban nem lehetséges a módosított jellemzőtérbeli vektorok ismerete nélkül. Lehetőségünk van viszont arra, hogy a $\tilde{\mathbf{K}}$ mátrixot $\mathbf{K}$-val kifejezzük.

Használjuk a következő jelöléseket: $K_{ij} = \left(\mathbf{\Phi}^T(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)\right)$, $\mathbf{1}_{ij} = 1$ minden i, j-re és $(1_N)_{ij} = 1/N$. Ezek után $\tilde{\mathbf{K}}$ számítása:

$$\begin{aligned}
\tilde{K}_{ij} &= \left(\left(\mathbf{\Phi}(\mathbf{x}_i) - \frac{1}{N}\sum_{p=1}^{N}\mathbf{\Phi}(\mathbf{x}_p)\right) \cdot \left(\mathbf{\Phi}(\mathbf{x}_j) - \frac{1}{N}\sum_{k=1}^{N}\mathbf{\Phi}(\mathbf{x}_k)\right)\right) \\
&= K_{ij} - \frac{1}{N}\sum_{p=1}^{N}1_{ip}K_{pj} - \frac{1}{N}\sum_{k=1}^{N}K_{ik}1_{kj} + \frac{1}{N^2}\sum_{p,k=1}^{N}1_{ip}K_{pk}1_{kj} \\
&= \left(\mathbf{K} - \mathbf{1}_N\mathbf{K} + \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N\right)_{ij}.
\end{aligned} \tag{2.36}$$

Most már kiszámíthatók a sajátértékek és a sajátvektorok, a főkomponensek számítása pedig ugyanaz, mint a nem központosított adatok esetében.

Jelvisszaállítás Mivel a kernel PCA a jellemzőtérben határoz meg főkomponenseket, ezért a főkomponensekből szintén a jel jellemzőtérbeli reprezentációját tudnánk előállítani. Viszont a kernel trükk miatt valójában nem is dolgozunk a jellemzőtérben, hiszen a jellemzőtérbeli vetületeket is meg tudjuk határozni a kerneltérbeli reprezentáció segítségével. Ha azt szeretnénk tudni, hogy mi a jellemzőtérbeli közelítő reprezentáció hatása a bemeneti térben, akkor a jellemzőtérbeli főkomponensekből vissza kell állítanunk a jelet a bemeneti térben. Ez a feladat egyáltalán nem triviális, sőt nem is feltétlenül egyértelmű. A visszaállításra Sebastian Mika [Mik99] és munkatársai javasoltak közvetett eljárást. E szerint a bemeneti térben keresünk olyan vektort, amelynek jellemzőtérbeli főkomponensei minél inkább hasonlóak a visszaállítandó adatok főkomponenseihez.

Jelöljük az eredeti adatok m főkomponens alapján kapott jellemzőtérbeli közelítő reprezentációját $\hat{\mathbf{X}}_m$-mel. Ekkor

$$\hat{\mathbf{X}}_m = \sum_{k=1}^{m} \beta_k \mathbf{V}^{(k)} \tag{2.37}$$

vagyis a közelítő reprezentáció a jellemzőtérbeli sajátvektorok lineáris kombinációjaként állítható elő. A visszaállításhoz olyan bemenetet keresünk, melynek a jellemzőtérbeli képe minél kisebb mértékben tér el $\hat{\mathbf{X}}_m$-től. E mögött az a feltevés áll, hogy ha két vektor jellemzőtérbeli reprezentációja között az eltérés kicsi, akkor a bemeneti térben is kicsi a köztük lévő eltérés. Négyzetes hibakritériumot alkalmazva ez azt jelenti, hogy keressük azt az $\hat{\mathbf{X}}$ bemeneti vektort, melyre

$$C\left(\hat{\mathbf{x}}\right) = \left\| \mathbf{\Phi}\left(\hat{\mathbf{x}}\right) - \hat{\mathbf{X}}_m \right\|^2 \tag{2.38}$$

minimális. Behelyettesítve (2.38)-be (2.37)-at és $\mathbf{V}(k)$ (2.23) összefüggését, az eltérésre a következőt kapjuk:

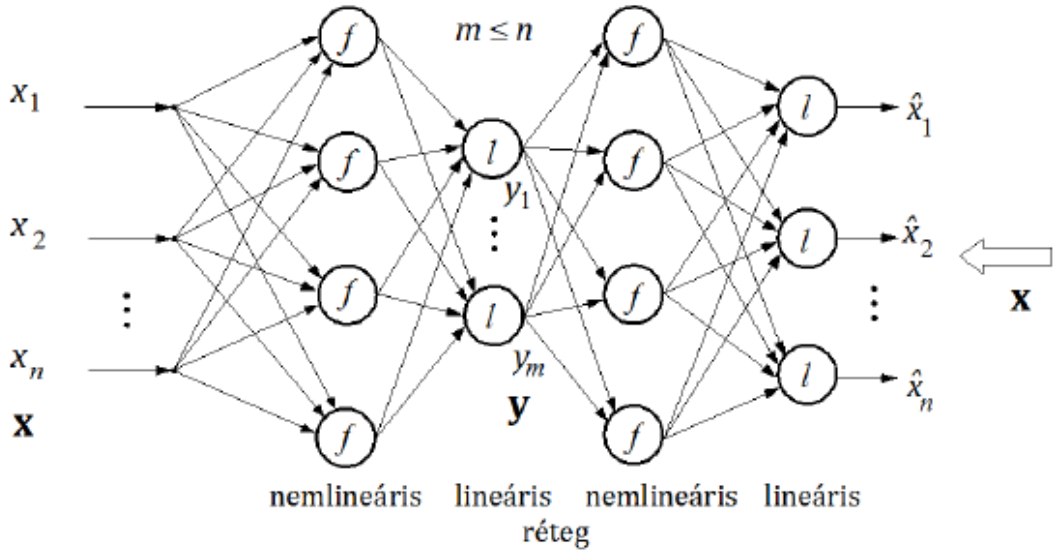
$$C\left(\hat{\mathbf{x}}\right) = K\left(\hat{\mathbf{x}}, \hat{\mathbf{x}}\right) - 2\sum_{k=1}^{m} \beta_k \sum_{i=1}^{P} \alpha_i^{(k)} K\left(\hat{\mathbf{x}}, \mathbf{x}_i\right) + \Omega, \tag{2.39}$$

ahol Ω független $\hat{\mathbf{x}}$-től. A (2.39) kritérium minimumát biztosító $\hat{\mathbf{x}}$ gradiens eljárással megkereshető, ha rögzítettük a kernel függvényt.

2.6.2. Nemlineáris altér algoritmusok

A lineáris altérfeladat autoasszociatív neuronhálós megoldásához hasonlóan a nemlineáris altérfeladat is megoldható többrétegű percetronnal. Ennél a megoldásnál azt használjuk ki, hogy egy többrétegű perceptron egy megfelelő méretű nemlineáris rejtett réteggel univerzális approximátor, vagyis tetszőleges folytonos leképezés közelítésére képes. Minthogy a tanítás a kívánt választól való átlagos négyzetes eltérés minimalizálását végzi most is, megfelelő többrétegű nemlineáris leképezésre alkalmas hálót a lineáris adattömörítő MLP-hez hasonlóan autóasszociatív módon tanítunk, várható, hogy a háló nemlineáris adattömörítésre képes lesz [Mal96]. A háló felépítése hasonló lesz a 2.2. ábrán bemutatott háló felépítéséhez azzal az eltéréssel, hogy most mind a transzformáció, mind a visszatranszformáció nemlineáris kell legyen, ami egy 5-rétegű hálózatot eredményez (2.5. ábra).

A tömörített reprezentációt ($\mathbf{y}$) a második rejtett réteg kimenetén nyerjük, míg a visszaállított adatokat a háló kimenetén. A nemlineáris adattömörítő MLP alapjában véve abban különbözik a lineáris tömörítést végző, a 2.2. ábrán bemutatott változattól, hogy kiegészül két nemlineáris rejtett réteggel, melyek alapvetően felelősek a nemlineáris leképezésért, és a tömörítést, illetve a visszaállítást hivatottak biztosítani. Az l-lel jelölt neuronoknál is alkalmazhatunk nemlineáris aktivációs függvényeket, bár a megfelelő működéshez erre valójában nincs szükség. A bemutatott nemlineáris hálózat – hasonlóan lineáris megfelelőjéhez – nem feltétlenül a nemlineáris főkomponenseket, hanem az m nemlineáris főkomponens által meghatározott altérbe eső vetületet határozza meg.



2.5. ábra. Nemlineáris altér MLP hálózat

2.7. Irodalom

[Bal89] Baldi, P. - Hornik, K. „Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima", *Neural Networks*, Vol. 2. No. 1. pp. 53-58. 1989.

[Bel57] Bellman R. E. „Dynamic programming", Princeton University Press. 1957.

[Dia96] Diamantaras, K. I .- Kung, S. Y. „Principal Component Neural Networks Theory and Applications", John Wiley and Sons, StateplaceNew York. 1996.

[Has89] Hastie, T. and Stuetzle, W. Principal curves, *Journal of the American Statistical Association* Vol. 84.pp. 502–516.

[Hor06] Altrichter M. – Horváth G. – Pataki B. – Strausz Gy. – Takács G. – Valyon J. (Horváth G szerk.) Neurális hálózatok, Panem, 2006.

[Hyv01] Hyvärinen, A. - Karhunen, J. - Oja, E. „Independent Component Analysis. John Wiley & Sons, New York, 2001.

[Mal96] Malthouse, E. C. „Some Theoretical Results on Nonlinear Principal Components Analysis", ftp from http://skew2.kellogg.nwu.edu/~ecm, 24 p. 1996.

[McL92] McLachlan, G. J. Discriminant Analysis and Statistical Pattern Recognition, Wiley, New York, 1992.

[Mik99] Mika, S. - Schölkopf, B. - Smola, A. - Müller, K-R. - Schultz, M. - Rätsch, G. „Kernel PCA and De-Noising in Feature Spaces", in: M. S. Kearns, S. A. Solla, D. A. Kohn (eds.) Advances in Neural Information Processing Systems, Vol. 11. pp. 536-542. Cambridge, MA. The MIT Press, 1999.

[Oja82] Oja, E. „A Simplified Neuron Model as a Principal Component Analyzer", *Journal of Mathematical Biology*, Vol. 15. pp. 267-273. 1982.

[Row00] Roweis, S. T. and Saul, L. K. (2000). Locally linear embedding, *Science,* Vol. 290. pp. 2323–2326.
[San89] Sanger, T. „Optimal Unsupervised Learning in a Single-layer Linear Feedforward Neural Network", *Neural Networks*, Vol. 2. No. 6. pp. 459-473. 1989.

3. fejezet

Ritka események detektálása

Az adatelemzés első fázisa általában az ún. *felderítő adatanalízis* (EDA = *Exploratory Data Analysis*) [114], [9]. Az EDA célja a mérési regisztrátumok mögötti tartalom felfedése és egy kezdeti, gyakran még nem formalizált modell megalkotása. Az EDA-t jellemzően a tárgyterület szakértője végzi, aki statisztikai tudása birtokában a mérési adatokat interpretálja és általánosítja.

A legtöbb mérés esetében természetes velejáró a statisztikai zaj, amely nem befolyásolhatja érdemben a modellalkotást. A műszaki gyakorlat *robusztus modelleket* igényel, amelyekből származtatott statisztikai jellemzőket a zaj nem, vagy csak kis mértékben befolyásolja.

A mérések során előfordulnak a többitől durván elkülönülő, jellegzetesen ritka *kilógó értékek* (*outlierek*). Ezek forrása lehet mérési hiba is, az ezekből származó kilógó értékeket azonosítás után figyelmen kívül kell hagyni. Számos alkalmazás esetében azonban ezek ritkán előforduló, különleges, a jó működéstől eltérő viselkedésre utaló, ún. *ritka események*, melyekre a helyes működés visszaállítása érdekében reagálni kell.

Ezen feladatok motiválták a ritka események detektálására és analízisére irányuló kutatásokat; ez napjainkra az orvosi diagnosztikában (daganatos sejtek, sejtmutációk detektálása) a rendszerüzemeltetésben (teljesítmény túlterhelődés, hálózati támadások felderítése), de például a pénzügyi szektorban (csalásfelderítés) is aktív kutatási területté vált.

A minőségileg eltérő viselkedést jelző adatpontok „*kilógónak*” címkézését és későbbi analízisét jellemzően szakértők végzik. Magas dimenziószámú és sok mintát tartalmazó adathalmazok elemzésében a kilógó értékek detektálásának és kategorizálásának algoritmikus támogatása elengedhetetlen. Különösen igaz ez a felderítő adatelemzésben, melyben a szakértő támogatást igényel az adatok megértéséhez és az alkalmazandó modellezési megközelítés kiválasztásához.

Ritka események jellemzően olyan problémák esetén fordulnak elő, ahol az adathalmaz nemcsak „széles”, hanem egyben a megfigyelések száma is különösen nagy, hiszen a ritka események markáns megjelenése éppen ritkaságuk miatt nagyszámú mintát feltételez.

Jelen fejezet a napjainkban ritka események felderítésére leggyakrabban használt megközelítéseket mutatja be.

3.1. Anomáliák és ritka események

A „*ritka események*" fogalmának nincs pontos definíciója. Jellemző, hogy kis előfordulási gyakoriságuk miatt számosságuk messze elmarad a normál pontokéhoz képest, manifesztációjuk jellege (nagyság, gyakoriság stb.) akár mintahalmazonként is változó lehet.

Intuitíven a ritka események jellemzése általános értelemben a kilógó értékekkel foglalkozó *anomáliadetektálás és -karakterizálás* egy alesete. Az, hogy az anomáliák közül mik adódnak mérési hibákból, illetve mik ritka eseményekből, csak a feltáró adatelemzés során, a hibás jelenségekkel való összekapcsolással dönthető el.

Az egységes tárgyalás érdekében így jelen alfejezetben előbb megadjuk az anomáliák (*outlierek, kivételek*) egy jellemző *osztályozását* [121] [18] [101], majd azok segítségével az anomáliák és a ritka események közötti *kapcsolatot* definiáljuk.

Az anomáliák egy lehetséges csoportosítása azon alapul, hogy a viselkedési eltérés valamely egyedi attribútumban is markánsan megmutatkozik-e, vagy csak a teljes állapotvektor elemzése mutatja-e ki.

Pontanomália – valamely egyedi attribútum értékkészletében *a szokásos tartományon kívül eső* adatpont.

Pontanomália például egy általában alacsony terheltségű processzor 99%-os terhelését reprezentáló adat.

A 3.1. ábrán például két nagyobb klaszter mellett az x_1 és x_2, valamint az X címkéjű csoportba tartozó pontok láthatólag kívül esnek minden nagyobb, a jellemző tartományokat reprezentáló klaszterből, így későbbi vizsgálatokat igényelnek.

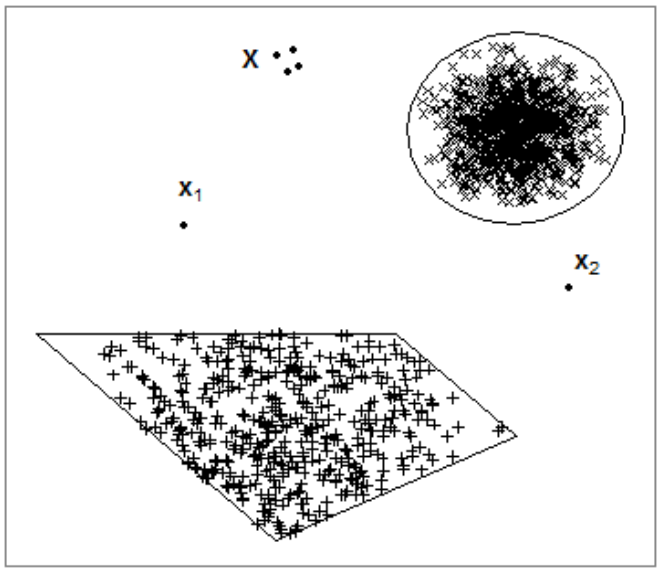
Kollektív anomália – a teljes adathalmazt tekintve a szokásos tartományon kívül eső adatpont,amelynek az egyes attribútumokra vetített értéke a normális értékek közé esik.

A kollektív anomáliákra a 3.2. ábrán látható EKG jel szabálytalan lefutása mutat példát: noha a -5,7-es érték a jel peremeloszlását tekintve nem tekinthető furcsának, az egymás után következő, hasonló értéket felvevő pontok halmaza már rendellenességet jelezhet.[1]

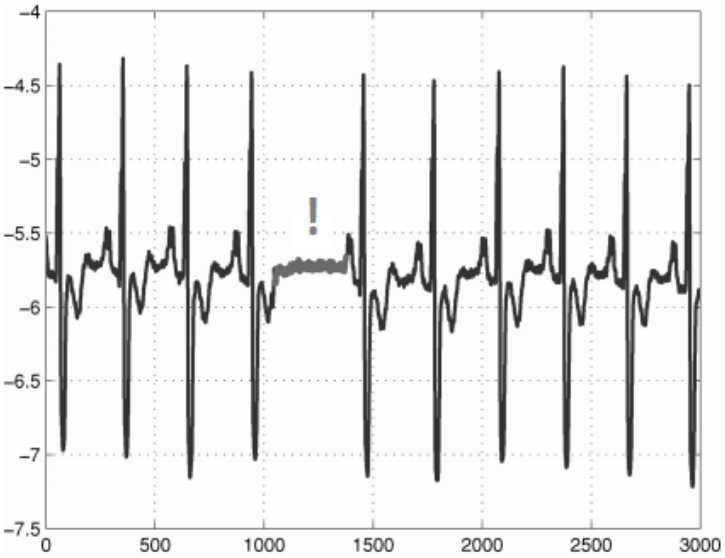
Miközben a műszaki gyakorlat szempontjából mindkét kategória fontos, detektálásuk és jellemzésük metodikailag alapvetően eltérő.

- A pontanomáliák észlelése és kezelése egyszerűbb, hiszen az érintett attribútum értéke könnyen észlelhetően a szokásos tartományon kívül esik.
- A kollektív anomáliák esetében azonban az egyes attribútumok értéke külön-külön a szokásos tartományon belül helyezkedik el, így észrevételük és jellemzésük mélyebb és átfogóbb vizsgálatot igényel.

[1] Az ábra forrása: [18], 9. oldal.



3.1. ábra. Példa pontanomáliára



3.2. ábra. Példa kollektív anomáliára

A kilógó értékeket detektáló algoritmusok értékelése jellemzően az alábbi általános paraméterek alapján történik:

Detektálási sikeresség: az események „normál” és ”ritka” kategóriákba sorolása már csak a statisztikai jelleg miatt is tévedhet. A detektálási sikeresség a helytelenül osztályozott kieső pontok értékelésével kapcsolatos preferenciánkat mutatja meg, azaz jelzi a hamis pozitív és hamis negatív értékelés költségét.

A ritka eseményeket általában utóvizsgálatnak szokás alávetni, így a gyanús eseményeket is inkább a ritka esemény jelöltekhez sorolják, de a hatékonyság érdekében (a hamis pozitívak számát ésszerű mértékre szorítandó) a normál tartomány határát lehetőség szerint pontosan kell meghatározni.

Komplexitás: különösen a nagyméretű, nagy dimenziószámú adathalmazok esetén kritikus az egyes algoritmusok idő- és tárkomplexitása. Az időbeli komplexitás különösen kritikus online analízisnél, ahol a rekordok beérkezése nagyon gyors lehet, míg offline elemzés során előfordulhat, hogy az adathalmaz egésze nem fér be a memóriába.

Bemeneti hangoló paraméterek: azt jelzik, hogy az adott algoritmus alkalmazásához mennyire szükséges az adatokat generáló folyamatok előzetes ismerete, illetve annak birtokában mennyire hatékonyan hangolható az algoritmus. A legtöbb detektáló algoritmus például a kilógó értékek arányával vagy a végső címkézést befolyásoló küszöbértékekkel kapcsolatban vár becslést.

Eloszlásvizsgálat: az algoritmusok különböznek az adatpontok eloszlásával kapcsolatos feltételezések mélységében:

- A paraméteres algoritmusok feltételezik, hogy ismert a vizsgálandó adatpontok eloszlása.
- A nemparaméteres algoritmusok az eloszlással kapcsolatban semmilyen feltételezéssel nem élnek.
- A félparaméteres (*semiparametric*) megközelítések ugyan nem határolják be az eloszlás típusát és paraméterezését (pl. exponenciális eloszlás $\lambda = 0.3$ paraméterrel), de bizonyos megkötéseket tehetnek arra vonatkozóan (pl. az eloszlás megfelelően „sima” legyen).

Kiválasztott elemek száma: iteratív algoritmusok konvergenciájának szempontjából kritikus, hogy az egyetlen adatpontból indul vagy már kezdetben is az adatpontok egy részhalmazát tekinti alapnak, hiszen ezek egy-egy lépés során meghatározzák a leginkább kilógó értékeket, majd ezeket feldolgozás után törlik az adathalmazból, vagy közelebbiekkel helyettesítik.

Lokalitás: mind a detektálási sikeresség, mind az algoritmus komplexitása szempontjából meghatározó, hogy egy-egy adatpont vizsgálatakor a teljes adathalmazt használja-e az algoritmus az adatpont besorolásához vagy csak az adatpont egy szűkebb környezetét.

A priori címkézés: adatok csoportosításánál jelentős segítség, ha néhány pontról biztosan tudjuk, hogy mely csoportba tartoznak.

- *A felügyelet nélküli* algoritmusok semmilyen előzetes címkézést nem igényelnek.
- *Felügyelt* esetről beszélünk azokban az esetekben, amikor mind a normál, mind a kilógó értékek tartományából legalább egy pontra ismerjük egy előzetes címkézés eredményét.

- *Félig felügyelt* esetben a két tartomány közül csak az egyikből ismerünk adatpontokat (leginkább egy előzetesen megjelölt normál tartomány ismerete jellemző).

3.2. Detektálási megközelítések

A ritka adatpontok felderítésének első lépése az attribútumtér jellemző tartományainak meghatározása, az adatok zöme ugyanis ide esik, megkönnyítve a statisztikai értelemben pontos határok meghúzását. A ritka pontok ezek után a normál tartományban látható, jellegzetesen csomósodó (klaszterbe foglalható) értékektől elkülönülőek lesznek.

A jó adatpontok tartományának meghatározása azonban *robusztus* módszereket igényel, amelyek biztosítják a szokásostól jelentősen eltérő adatpontok érdemi befolyásmentességét.

A normál tartományt kijelölő klaszterek határai azonban a statisztikai szórás miatt nem élesek. Ezért egy-egy pontnál eldöntendő, hogy az kilógó esemény-e vagy ugyan a jó tartományhoz tartozik, de a statisztikai szórás miatt a klaszter szélén helyezkedik el.

Döntési kritériumként két megközelítés szokásos.

A távolság alapú technikák a kilógó tulajdonságra fókuszálva azt vizsgálják, hogy az adott adatpont a jellegzetes mérési értékeket tartalmazó klaszterektől távol van-e.

A sűrűség alapú technikák a ritka események kis előfordulási gyakoriságát alapul véve azt vizsgálják, hogy az adatpont a sűrűn előforduló mérési értékek között van-e.

3.2.1. Távolság alapú módszerek

A távolság alapú módszerek feltételezik, hogy a különleges pontok a többségtől vagy a teljes adathalmaz „középpontjától" messze vannak:

Pontanomáliák esetében ez *egyetlen attribútum* vizsgálatával eldönthető, hiszen a kilógó érték kiesik a szokásos értékeket tartalmazó intervallumból.

Kollektív anomáliák esetében ugyan elvben az *összes attribútumot* vizsgálni kell, de gyakran az anomália jelleg detektálható az attribútumok egy szűkebb részhalmazát elemezve is.

Ehhez jellemzően előbb a *normál tartományt megadó klaszterek* és az azokat összetartó, illetve azoktól szeparáló jó attribútumok halmazának kiválasztása válik szükségessé. Így a távolságok kiszámítását ún. *relevanciaszűrés*, azaz az egyes klaszterek koherenciafaktorainak meghatározása előzi meg.

A kiinduló ötlet egyszerűsége dacára a távolság fogalmának definíciója kritikus többdimenziós adatok esetében, hiszen több különböző állapotvektor komponenst kell összehasonlítani.

A mérési adatvektorok különböző komponensei különböző fizikai mennyiségekre vonatkozhatnak, sőt az egyes komponensek értéke az alkalmazott mértékegységtől függően is változik.

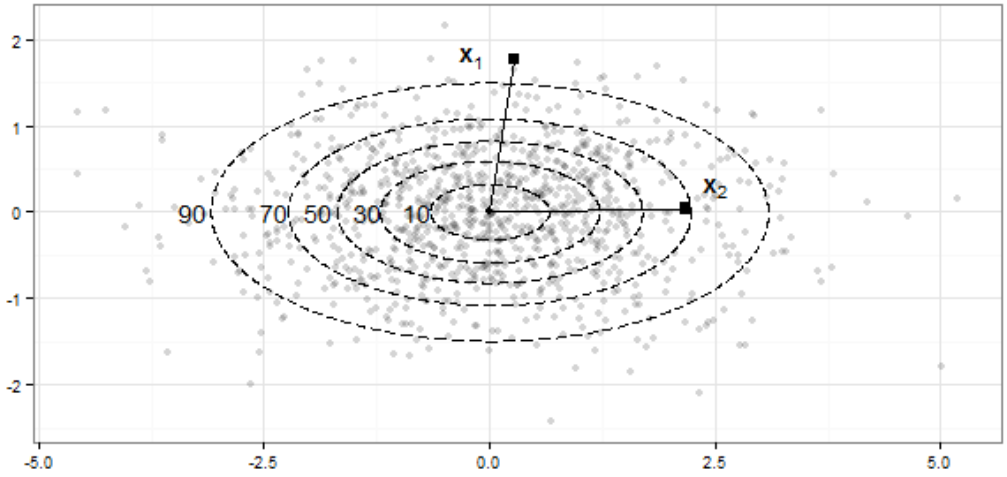
Ezért a két, x_1 és x_2 pont között értelmezett $\|x_2 - x_1\|$ klasszikus euklideszi távolság (és általában a Minkowski-távolság) előzetes normálás nélkül általában nem alkalmazható: a valós együtthatókkal szorzásra érzéketlen, ún. *skálainvariáns* függvények használata válik szükségessé.

Ezek egyike a dimenziók együttes eloszlásfüggvényét figyelembe vevő, ún. *Mahalanobis távolság*. Ez képes a különböző dimenziók egységes és egybevethető keretbe foglalására. Mivel jelen alfejezet több, Mahalanobis távolságot alkalmazó algoritmust is bemutat, a következő szakasz ennek formális definícióját és intuitív interpretációját mutatja be röviden.

A Mahalanobis távolság

A 3.3. ábrán egy kétdimenziós normál eloszlás látható, rajta a 10., 30., ..., 90. percentilist jelző kontúrvonalakkal. A várható értékhez legközelebb eső 10%-os ellipszisben a legsűrűbbek a pontok, a legkülső, 90%-oson kívül pedig a legritkábbak.

A Mahalanobis távolság azt mutatja, hogy a közelebbi kontúrvonalon belüli x_2 pont várhatóan nagyobb valószínűséggel tartozik a várható érték körüli csomósodáshoz, mint a távolabbi kontúrvonalon elhelyezkedő x_1, noha a két pontnak a középponttól vett euklideszi távolsága pont fordított nagyságú.



3.3. ábra. Mahalanobis távolság

Két, azonos eloszláshoz tartozónak feltételezett x_1 és x_2 vektor Mahalanobis távolságán az alábbit értjük:

$$D_{Mahal}(x_1, x_2) = \sqrt{(x_1 - x_2)^T S^{-1} (x_1 - x_2)}, \tag{3.1}$$

ahol S a közös eloszlás *kovariancia mátrixát* jelöli.

Hasonlóan, egy x_1 pont és egy M ponthalmaz Mahalanobis távolsága az alábbi:

$$D_{Mahal}(x_1, M) = \sqrt{(x_1 - \nu)^T S^{-1}(x_1 - \nu)}, \tag{3.2}$$

ahol ν az M halmaz „súlypontját" (általában átlagát), S pedig M kovariancia mátrixát jelöli.

A a 3.3. ábrán például az egyes kontúrvonalakra eső pontoknak a középponttól mért Mahalanobis távolsága azonos.

Korrelálatlan esetben az attribútumonkénti normálás után a kovariancia mátrix egységmátrix (az ellipszisek körré egyszerűsödnek), így a Mahalanobis távolság azonos az euklideszivel.

A 3.1 és 3.2 képlet által végzett távolságszámítások azonban az alapesetben érzékenyek a kilógó értékekre, hiszen azok a szóráshoz jelentősen hozzájárulhatnak. Ezért a távolság robusztus számításához az ahhoz szükséges középpont és a kovariancia mátrix robusztus becslése szükséges.

A befoglaló burok módszere

Tukey 1977-ben bemutatott anomáliadetektáló módszere [113] a folytonos távolságot ordinális értékekkel közelíti. Alapötlete, hogy a ponthalmazt hagymahéjszerűen beburkolva a mélyebb héjakon található pontok közelebb vannak a centrumhoz, azaz a pontot tartalmazó hagymahéj kívülről számított sorszáma egy távolságbecslés. Ezt a sorszámot a pont *féltér-mélységének* (*halfspace depth*) nevezzük.

A kilógó pontok tehát a kis sorszámú külső héjakon ritkásan helyezkednek el.

A Tukey-féle anomáliadetektáló módszer a féltér-mélységkiszámításához meghatározza a befoglaló poliédert, az általa tartalmazott pontokat „1" címkével jelöli, majd törli a halmazból, a megmaradt pontokra pedig ciklikusan folytatja a „2", „3", ... konvex burkok építését, a pontok címkézését majd törlését (3.4a. ábra).

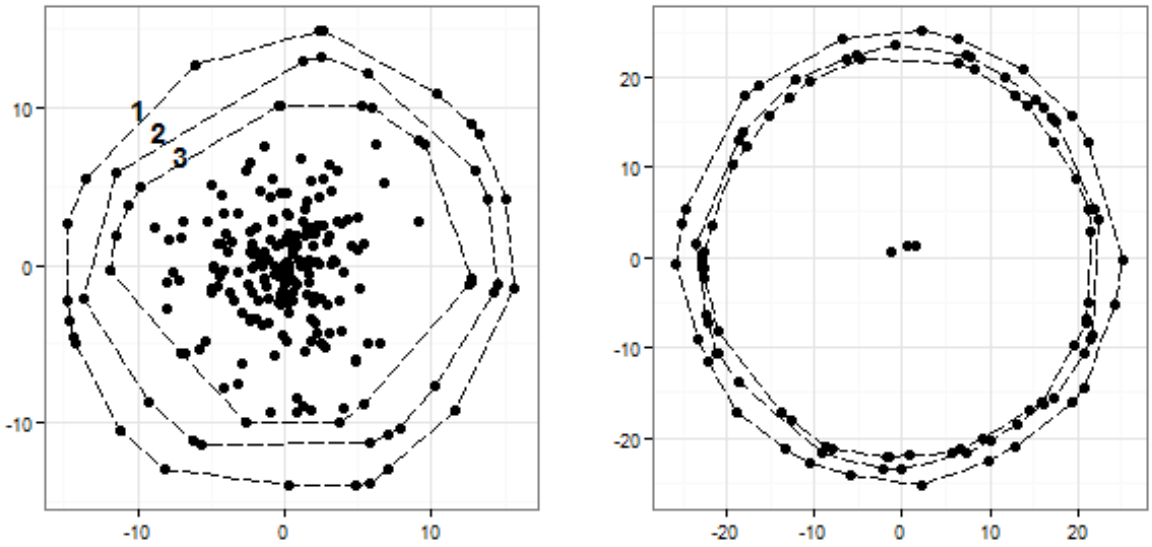
A Tukey-módszer alkalmazása azt feltételezi, hogy a helyes pontok a mérési pontfelhő geometriai közepe környékén helyezkednek el. Abban az esetben viszont, ha az a feltételezés hamis és a helyes értékek a középponttól távol egy gyűrűszerű sávban vannak, a Tukey-módszer hibásan a távol lévő helyes értékeket jelöli kilógónak (3.4b. ábra).

- *Egyváltozós* $\underline{X} = \{x_1, x_2, \ldots, x_n\}$ adatkészlet esetében az x_i pont féltér-mélysége az x_i-nél nem nagyobb és x_i-nél nem kisebb elemek számának minimuma (3.5. ábra):

$$depth(x_i) = \min\{|\{j : x_j \leq x_i\}|, |\{j : x_j \geq x_i|\}\}. \tag{3.3}$$

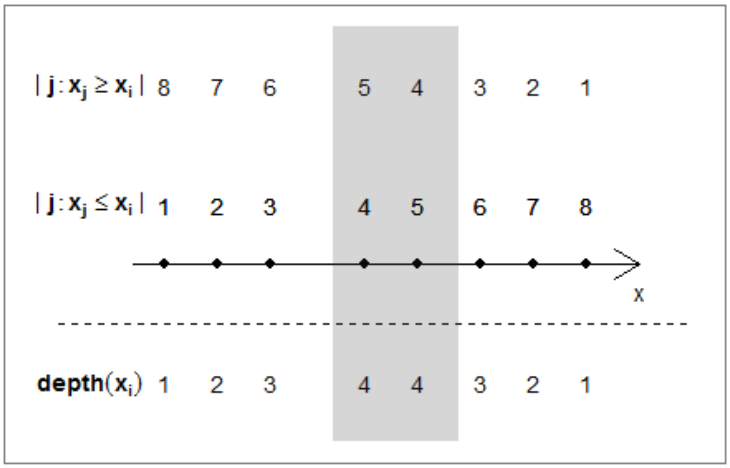
  Így az adathalmaz szélén a minimumra és a maximumra a féltér-mélység „1", míg a középpont szerepét játszó mediáné maximális.

- *Többváltozós* esetben egy x pont féltér-mélységén az összes lehetséges egydimenziós vetítésnél megkapott féltér-mélységének minimumát értjük: ekkor általános helyzetű bemeneti ponthalmaz esetén az „1" féltér-mélységű pontok a ponthalmaz geometriai



(a) Kilógó pontok kívül (b) Kilógó pontok belül

3.4. ábra. Féltér-mélység meghatározása kétdimenziós esetben



3.5. ábra. Féltér-mélység meghatározása egydimenziós esetben. A legnagyobb féltér-mélységű adatpontok középen

értelemben konvex burkát határozzák meg, a „2” féltér-mélységű pontok az „1” féltér-mélységűek elhagyásával keletkező halmaz konvex burkát stb.

A nagy féltér-mélységű pontok egyre inkább a „felhő belsejében” vannak. Így a maximális értékkel rendelkező utolsó pontok az adathalmaz mediánjának, a kis értékű pontok pedig anomáliának tekinthetőek (3.4a. ábra).

Az algoritmus előnye a robusztussága, hiszen egy-egy kieső pont a környezetén kívüli normális pontok metrikáját alig befolyásolja.

A Tukey-féle féltér-mélység magas dimenziójú adatokon való számítására hatékony algoritmus az **ISODEPTH** [97], amelynek egy R [91] nyelvű implementációja a *depth* [83] csomagban is megtalálható.

A DB algoritmus

A DB (*Distance Based*) algoritmus [71] alapgondolata a befoglaló burokéhoz hasonló: a többségi pontoktól messze eső, kívülálló pontoknak kevesebb szomszédjuk van.

A kieső pontokat a DB algoritmus úgy definiálja, hogy egy előre megadott lokális környezetükben a pontsűrűség kisebb, mint a felhasználó által megadott küszöbérték. A számításhoz a felhasználó a vizsgálandó környezet nagyságát egy ε sugarú hipergömbként definiálja π elvárt szomszédossági aránnyal, amely az n mérési pontból egy hipergömbbe esők relatív gyakorisága.

Az anomáliákat gyűjtő $OutlierSet(\varepsilon, \pi)$ halmaz ezek után:

$$OutlierSet(\varepsilon, \pi) = \{x_i : neighbour.ratio(x_i) \leq \pi\}, \tag{3.4}$$

ahol

$$neighbour.ratio(x_i) = \frac{|\{x_j : \|x_j - x_i\| \ \leq \varepsilon\}|}{n}. \tag{3.5}$$

A *neighbour.ratio* érték kiszámítását az R környezetben a *fields* csomag [53] **fields.rdist.near** függvénye végzi.

Szóráscsökkenésen alapuló algoritmusok

A gyakori értékeket befoglaló, sűrű csomóktól távol fekvő kieső pontok elhagyása csökkenti a mintahalmaz szórását. Többdimenziós esetben ennek egy jellemzője az ún. *általánosított szórás*, amely a kovariancia mátrix determinánsának értéke. A determináns nagysága meghatározza a befoglaló hiperellipszoid térfogatát.

A 90-es évek két meghatározó megközelítése az **MCD** (*Minimum Covariance Determinant*) [94] és **MVE** (*Minimum Volume Ellipsoid*) [95] algoritmusok voltak. Mindkettő célja az adathalmaz egy „összetartozó” részhalmazának megtalálása, majd azok alapján a teljes halmaz robusztus középpontjának és kovariancia mátrixának meghatározása.

- Az MCD ehhez azt a h méretű ponthalmazt határozza meg, amelynek kovariancia mátrixának determinánsa az összes ilyen méretű ponthalmaz közül a legkisebb.
- Az MVE a legkisebb térfogatú, legalább h pontot tartalmazó hiperellipszoidot keresi.

Mindkét esetben a h a felhasználó által megadott, a legnagyobb jó értékeket tartalmazó halmaz méretét becslő küszöbparaméter. Ennek meghatározása egyszerű akkor, ha a jó

értékek egyetlen csomóban helyezkednek el és ismert a kieső elemek előfordulási gyakorisága. Például 1%-nyi kilógó adat esetében $h = \lceil n \cdot (1 - 0.01) \rceil$ környezetében érdemes induló értéket választani.

Az algoritmusok eredetileg kimerítő keresést alkalmaztak, amelyek nagyméretű és magas dimenziójú adathalmazokon való alkalmazása már 1000-es nagyságrendi számosságú adathalmaznál is korlátokba ütközött, ez iteratív, közelítő algoritmusok létrehozását inspirálta.

Az összetartozó részhalmazok iteratív keresése egy kezdeti adathalmazból kiindulva minden lépésben hozzáveszi ahhoz a kohéziós metrika szerint közeli pontokat és elhagyja a módosított halmaztól egy küszöbértéknél távolabb kerülőket (Algoritmus 1).

Az egyes specializált algoritmusok az általános kereten belül az iterációk során az aktuális közelítő H_j halmaz környezetében megvizsgált ponthalmaz $\Lambda_{selection}$ kiválasztási kritériumában, az alkalmazott közelséget értékelő $\delta_{proximity}$ metrikában, illetve a $\Lambda_{termination}$ megállási kritériumban térnek el.

Algoritmus 1 Szóráscsökkenésen alapuló iteratív megközelítés általános sémája

Input: X: bemenő adatpontok halmaza
$\Lambda_{termination}$: algoritmusfüggő megállási kritérium
$\Lambda_{selection}$: kiválasztási kritérium
$\delta_{proximity}$: közelség értékelése
Output: $X_{out} \subseteq X$: normál ponthalmaz becslése

1. $H_0 :=$ kiinduló részhalmaz
2. $j := 0$
3. $n := |X|$
4. **while** $\Lambda_{termination}$ **do**
5. $Model_{H_j} :=$: az aktuális részhalmaz „modellje”, a későbbi távolságszámítás alapja
6. $H_0 := \emptyset$
7. **for** $i := 1$ **to** n **do**
8. $d_i := \delta_{proximity}(x_i, Model_{H_j})$
9. **if** $\Lambda_{selection}(d_i)$ **then**
10. $H_{j+1} := H_{j+1} \cup \{x_i\}$
11. **end if**
12. **end for**
13. $j := j + 1$
14. **end while**
15. $F_{out} := H_j$

A fenti sémára illeszkedő algoritmusokra példák a **FAST-MCD** (*Fast Minimum Covariance Determinant*) [94] és a **BACON** (*Blocked Adaptive Computationally Efficient Outlier Nominators*) [11]. Mindkét algoritmus az iteráció során az aktuális adathalmazt a Mahalanobis távolság alapján fokozatosan módosítja.

FAST-MCD

A megközelítés lényege, hogy a soron következő részhalmazt az aktuálishoz legközelebbi h pont alkossa. Az algoritmus megáll, ha az új részhalmaz „stabil”, azaz az

azt megelőzővel megegyezik. Az algoritmus konkrét megvalósítását a 3.1. táblázat foglalja össze.

3.1. táblázat. FAST-MCD megvalósítás

H_0	véletlenszerűen választott részhalmaz
$Model_{H_j}$	$(\nu(H_j), S(H_j))$ pár, ahol $\nu(H_j)$ a részhalmaz súlypontja, $S(H_j)$ pedig a kovariancia mátrixa
$\delta_{proximity}(x_i, M)$	x_i M-től vett Mahalanobis távolsága
$\Lambda_{selection}$	h darab legkisebb távolságú elem kiválasztása
$\Lambda_{termination}$	a részhalmaz stabil: $H_{j+1} = H_j$

[94] igazolja, hogy a determináns értéke az egyes iterációkban csökken, mivel a fenti megvalósításban teljesül a $det(H_{j+1}) \leq det(H_j)$ feltétel. A kezdőhalmaz véletlenszerű kiválasztásának következményeit a tapasztalatok szerint kiküszöböli a kellően nagy iterációs szám (a tapasztalat szerint az adathalmaz számossága már alkalmas választás lehet), az egyes iterációkban visszaadott ponthalmazok közül az abszolút minimumú determinánsút választva.

BACON

Előfeltétele, hogy a szakértő által bemenetként megadott kiinduló részhalmaz kilógó értékektől mentes legyen.

Az egyes lépésekben előállított új részhalmaz azokat a pontokat tartalmazza, amelyeknek az előző részhalmaztól vett távolsága a felhasználó által bemenetként megadott küszöbérték alatt van. Az algoritmus akkor áll meg, ha az új részhalmaz mérete kisebb lenne az előzőénél. A BACON algoritmus konkrét megvalósítását a 3.2. táblázat foglalja össze.

3.2. táblázat. BACON megvalósítás

H_0	szakértő által meghatározott, a feltételezés szerint anomáliamentes részhalmaz
$Model_{H_j}$	$(\nu(H_j), S(H_j))$ pár, ahol $\nu(H_j)$ a részhalmaz súlypontja, $S(H_j)$ pedig a kovariancia mátrixa
$\delta_{proximity}(x_i, M)$	x_i M-től vett Mahalanobis távolsága
$\Lambda_{selection}$	d_i egy előre meghatározott küszöbérték alatti
$\Lambda_{termination}$	a részhalmaz biztonságosan már nem bővíthető: $\lvert H_{j+1}\rvert < \lvert H_j\rvert$

Mindkét algoritmus a jó értékekből álló halmaz kompakt és robusztus reprezentációját keresi. Az MCD biztosítja a legalább h méretű végeredményhalmazt. Ezt a BACON nem

garantálja, hiszen ha a ponthalmazban gyűrűszerű, a küszöbértéket meghaladó lyukak vannak, akkor az algoritmus akkor is leáll, ha a gyűrűn kívül még lennének jó tartománybeli elemek.

Mindezzel együtt a BACON algoritmus jellemzően kisebb erőforrás igényű, hiszen a $\Lambda_{selection}$ kritérium kiértékelése csak egy konstans értékkel való összehasonlítás az MCD-beli h darab legkisebb távolságú adatpont kiválasztása helyett.

A BACON algoritmust az előzetes vizuális analízis mind a kezdeti halmaz kiválasztásában, mind pedig a küszöbérték meghatározásában alapvetően támogatja.

A fent tárgyalt algoritmusok a továbbiakban számos finomított megoldást ihlettek, az eredeti megközelítések egy implementációját az R környezet *robustBase* [82], illetve *robustX* [105] csomagjai tartalmazzák.

A PALM algoritmus

A félparaméteres **PALM** (*Partial Augmented Lagrangian Method*) algoritmus [61] arra a speciális, de gyakran teljesülő feltételezésre épít, hogy a ritka eseményeket egy közös ok hozza létre. Ekkor a ritka események maguk is csomósodnak, sőt a jó és ritka események csomói néhány koordinátájuk alapján megkülönböztethetőek. Ezek alkotják az ún. *szeparáló attribútumhalmazt*.

Ez a *kompaktsági feltétel* azt mondja ki, hogy a ritka események adatpontjainak egymáshoz képesti távolsága kisebb a normál pontokhoz képestinél, legalábbis a ritka és jó pontokat szeparáló attribútum részhalmazra vetítve. Így a feladat a legkisebb össztávolságot produkáló, a felhasználó által bemenő paraméterként megadott méretű pont- és attribútumhalmaz meghatározása.

Az algoritmus a ritka pontok r valószínűségét és a szeparáló attribútumok becsült l számát várja bemenő paraméterként (azaz az algoritmus nem igényli a közös okot kifejező attribútumok előzetes konkrét meghatározását, csak egy becslést azok számára).

Jelölje az a bináris vektor a ritka jelenség csomópontjainak tagsági függvényét: a_i értéke 1, ha az i-edik adatpont ritka és 0, ha a jó tartományba sorolt. Ekkor n méretű adathalmaz esetén az $\widehat{r} = \sum_{i=1}^{n} a_i/n$ egy becslője a ritka esemény valószínűségnek. A nehézséget épp az okozza, hogy r szokásosan egy kis érték és a gyakoriság kellően pontos becsléséhez nagy adathalmazra van szükség.

Hasonlóan legyen a b bináris vektor a szeparáló attribútum részhalmaz tagsági függvénye, azaz b_j értéke 1, ha a j-edik attribútum szeparáló! Ekkor összesen l attribútumot kell kiválasztanunk: p dimenziójú állapotvektor mellett $\sum_{j=1}^{p} b_j = l$ adódik.

Ha x_i^j az i. adatpont j. attribútumának értéke, akkor a kompaktsági feltétel alapján a ritka adatok a kiválasztott attribútumhalmazra vetített, minimális össztávolságú alakzat

adatpontjai lesznek:

$$\min_{a,b} \frac{1}{nr} \sum_{i=1}^{n} \sum_{k=1}^{n} a_i a_k (\sum_{j=1}^{p} b_j (x_i^j - x_k^j)^2) \tag{3.6}$$

$$\begin{aligned} \sum_{i=1}^{n} a_i &= nr, \quad a_i \in \{0,1\} \\ \sum_{j=1}^{p} b_j &= l, \quad b_j \in \{0,1\}. \end{aligned}$$

A **PALM** algoritmus [61] a kitüntetett attribútumok illetve a ritkáknak tekintett minták felett iteratív megoldást kínál a fenti minimum meghatározására.

Előbb egy kezdeti attribútumhalmaz segítségével meghatározza az egymáshoz legközelebbi nr számú pontból álló ritka esemény csomó jelöltet, majd az ezt és a többi, jónak tekintett pontokat szeparáló attribútumokat. Ez az attribútumhalmaz egy új csomót határoz meg, majd következik az új részhalmaz számítás, és így tovább.

A módszer korlátozott érvényessége dacára jó kiegészítője a vizuális adatfeltárásnak, hiszen ott a szeparáló attribútumok ránézésre valószínűsíthetőek, azaz jó induló attribútumhalmaz adható.

3.2.2. Sűrűség alapú módszerek

A sűrűség alapú módszerek alapgondolata, hogy egy pont kívülállóságának mértékét annak lokális környezetétől való különbözősége határozza meg.

A 3.6. ábrán[2] a távolság alapú DB algoritmus egy szintetikusan előállított bemenete látható. A Cl_1 és Cl_2 klaszterek a jó működéshez tartozó, eltérő sűrűségű állapottérrészek, az x_1 és x_2 pedig kilógó pontok. Az x_2 pont közel esik a Cl_2 klaszterhez, ellenben a környezetébe eső szomszédok számát tekintve a Cl_1-ben lévőkhöz hasonló.

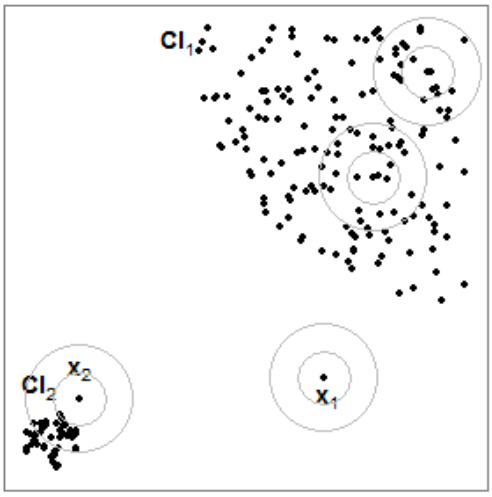
A távolság alapú DB algoritmus korlátos képességeit mutatja, hogy nem paraméterezhető úgy, hogy felismerje x_2 kilógó voltát és egyidejűleg Cl_1 adatpontjait a jó tartományba sorolja.

A távolság alapú módszerekre általában jellemző probléma egy lehetséges megoldásaként terjedtek el a *sűrűség alapú* anomáliadetektáló módszerek, amelyek a kategorizálandó adatpontnak a környezetében fekvő szomszédaihoz viszonyított, ún. *lokális sűrűségét* vizsgálják.

A LOF algoritmuscsalád

A **LOF** (*Local Outlier Factor*) megközelítés arra épít, hogy a jó tartományba tartozó csomópontok sűrűsége általában nagyobb, mint a ritka események csomóinak. Azokat a

[2] Az ábra forrása: [13], 2. oldal



3.6. ábra. A sűrűség alapú módszerek motivációja

pontokat tekinti kiugrónak, amelyeknek lokális sűrűsége kisebb, mint a szomszédaiké. Ezt a sajátosságot az ún. *lokális elérési sűrűség* ($lrd_k(x)$, *local reachability density*) és a *lokális kiugrás mértéke* ($lof_k(x)$, *local outlier factor*) értékeivel fejezi ki.

A LOF algoritmus a vizsgált x pont k legközelebbi szomszédjából álló $N_k(x)$ környezetébe eső valamennyi pontra távolságként egységesen a környezet legtávolabbi pontjától vett $kd(x)$ euklideszi távolságát használja (szemléletetesen a $kd(x)$ sugarú hipergömb belsejében fekvő pontokat a hipergömb felületére vetítjük). Az e környezeten kívül eső pontoknál a mérték a szokásos $\|x_1 - x_2\|$ euklideszi távolság.

A k vizsgálandó szomszédsági szám paramétert a felhasználó definiálja, ezzel képes a távolságok „simításának" finomhangolására. $k = 1$ esetében a legközelebbi szomszéd euklideszi távolsága alapján számol az algoritmus, $k = \infty$ esetében pedig a teljes adathalmaz legtávolabbi pontja szabja meg a működést.

Az x_2 pont x_1-től való $rd_k(x_2, x_1)$ *elérési távolsága* (*reachability distance*) formálisan tehát:

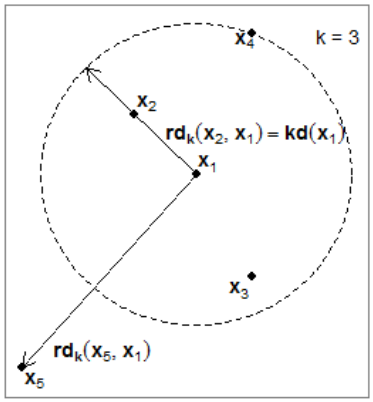
$$rd_k(x_2, x_1) = \begin{cases} \|x_1 - x_2\| & \text{ha } x_2 \notin N_k(x_1) \\ kd(x_1) & \text{ha } x_2 \in N_k(x_1) \end{cases} \tag{3.7}$$

A 3.7. ábra példáján $N_k(x_1) = \{x_2, x_3, x_4\}$ ($k = 3$). Miután x_4 a legtávolabbi pont, $kd(x_1) = \|x_1 - x_4\|$ és ugyanez az értéke $rd(x_2, x_1)$ és $rd(x_3, x_1)$-nek is.

Az $N_k(x_1)$-en kívüli x_5 pontnak a távolsága x_1-től a szokásos $\|x_5 - x_1\|$ euklideszi távolság.

Ezek után egy x pont *lokális elérési sűrűsége* annak k méretű környezetében ($N_k(x)$) a szomszédaitól számított elérési távolságok átlagának reciproka:

$$lrd_k(x) = 1/\frac{\sum_{x_i \in N_k(x)} rd_k(x, x_i)}{|N_k(x)|}. \tag{3.8}$$



3.7. ábra. Az x_1 pont elérhetőségi távolságának számítása

(Feltesszük, hogy az adathalmaz nem tartalmaz duplikátumokat, ellenkező esetben k azonos elem előfordulása is lehetővé válna, amelyek így egymás k környezetét kitöltenék. Ebből következne, hogy $lrd_k(x, x_i) = 0$ adódna $\forall i$-re, így az $lrd_k(x)$ értéke végtelen is lehetne.)

A 3.8. képletben a nevezőben a környezet mérete $|N_k(x)|$ szerepel, ez azokban az esetekben lehet k-nál nagyobb, ha a $kd(x)$ távolság, mint sugár által meghatározott hipergömb felületén egynél több pont található.

Az x pont *lokális kiugrásának mértékét* ekkor annak környezetébe tartozó pontok és a saját átlagos elérési sűrűsége hányadosának átlaga adja.

$$lof_k(x) = \frac{\sum_{x_i \in N_k(x)} lrd_k(x_i)}{|N_k(x)| \cdot lrd_k(x)}. \tag{3.9}$$

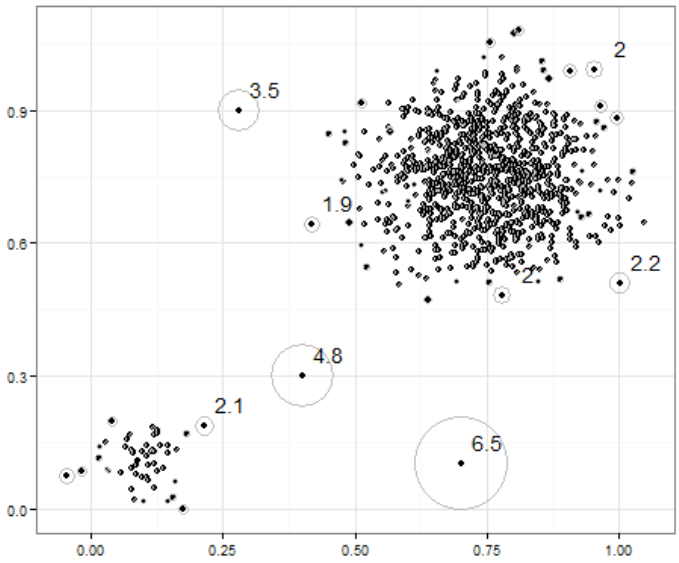
Egy sűrű klaszter belsejében ez az érték ≈ 1. A pont nagy kiugrási értékű, ha saját lokális elérési sűrűsége kicsi, a szomszédaié viszont nagy.

A 3.8. ábra két, különböző kovariancia mátrixú, kétdimenziós normál eloszlású szintetikus adatkészleten indított futás végeredményét mutatja. A csomópontok köré rajzolt sugár az adott csúcs lokális kiugrásának mértékét reprezentálja, az ábrán számérték is jelzi a legnagyobbakat.

Látható, hogy a klaszterektől messze fekvő pontok még nagy értékeket kaptak (így pl. a DB algoritmus által is detektált pontokat a LOF is kiemeli). A különböző sűrűségű klaszterek peremén azonban a pontok lokális kiugrásának mértéke hasonló, a klaszterek méretétől és kovariancia mátrixától független, 2.1 körüli lett.

A LOF algoritmus egy implementációja megtalálható az R környezet DMwR (*DataMiningWithR*) [111] csomagjában.

A LOF algoritmus finomított változatai annak távolságmértékét, illetve a környezet definícióját definiálják felül, előbbire a **COF** (*Connectivity-based Outlier Factor*), míg utóbbira a **LOCI** (*Local Outlier Correlation Integral*) algoritmus.



3.8. ábra. A LOF algoritmus működése. A pontok köré rajzolt kör sugara arányos a hozzájuk tartozó LOF értékkel

A COF algoritmus [107] az x pont k legközelebbi szomszédját iteratívan határozza meg: előbb hozzáveszi a környezethez az x pont legközelebbi szomszédját, majd az így kialakult részhalmazhoz keresi a legközelebbi pontot (a pont-halmaz távolság meghatározására a különböző implementációkban általában az egyes halmazbeli pontoktól vett távolságok minimumát, maximumát vagy átlagát használják), egészen addig, amíg a szomszédok halmazának számossága el nem éri az előre megadott k küszöbértéket.

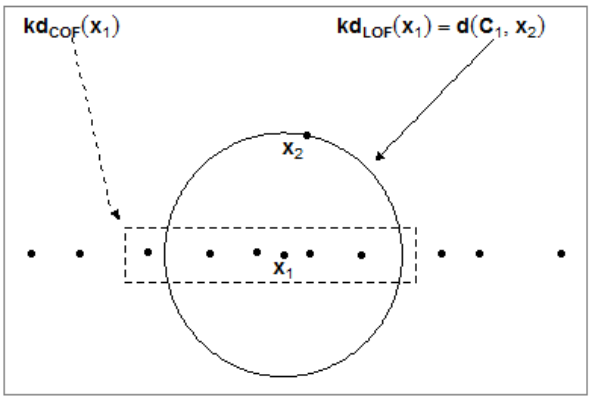
A távolságdefiníciók különbségére a 3.9. ábra mutat példát, itt $k = 5$, a pont-halmaz távolság pedig a minimumok alapján lett számítva.

A sűrűség alapú megközelítések egy csoportja, például a LOCI algoritmus [88], a pontok környezetét nem a szomszédok számából származtatja, hanem egy ε sugarú hipergömböt tekint szomszédságnak. Ez a megközelítés jobb eredményeket szolgáltat abban az esetben, ha a ritka események nagyon szétszórva helyezkednek el és esetleg keverednek a statisztikai szórás miatt kieső normál értékekkel.

Az NNDB algoritmus

A fent ismertetett algoritmusok viselkedését alapvetően befolyásolja az, hogy a szomszédossági mértékeket (k, ε) a felhasználó mennyire jól képes megbecsülni. Ezeknek a bemeneteknek a meghatározása gyakran nehézkes és a félreparametrizálás hamis statisztikai modelleket eredményez. Ennek kiküszöbölésére egy lehetséges megközelítés az, hogy az algoritmus a vizsgálandó lokális környezet nagyságát a futás során a mintahalmazhoz illeszkedve automatikusan határozza meg.

Az NNDB (*Nearest-Neighbor-Based Rare Category Detection for the Binary Case*) al-



3.9. ábra. A LOF és a COF megközelítések által használt távolságfogalom különbsége, $k = 5$ környezetmérettel

goritmus az adatpontok legnagyobb sűrűségváltásait keresi, hiszen ez jellemző az adatcsomó határpontjaira. Így a környezetéhez képest nagy szomszédossági mérőszámú pontokat jelöli meg, ezzel elősegítve a PALM algoritmus esetén ismertetett csomósodó részhalmazok detektálását.

Az NNDB félig felügyelt iteratív algoritmus: az egyes lépések során megkeresi azokat a pontokat, amelyek környezetében a sűrűségváltás a lehető legnagyobb, majd egy (szakértői tudással bíró) orákulumnak továbbítja sejtését. Amennyiben az orákulum gyakorinak minősíti a küldött pontot, úgy megnöveli a kör sugarát és a következő lépésben nagyobb környezetet vizsgál; ellenkező esetben megáll, hiszen talált egy szakértő által validált kilógó adatpontot.

A futáshoz szükséges kezdő sugarat az NNDB a ritka pontok (sejtett) számából számítja. Ha a ritka pontokra teljesül a kompaktsági feltétel, akkor a legnagyobb lokális sűrűségű pont éppen a ritkák közelében van. Így az első lépésben használt ε_1 sugárra a legkisebb olyan távolság adódik, amelyre még létezik olyan pont, amelynek a befoglaló hipergömbjében $n \cdot r$ pont található. A megközelítés pszeudo kódját az Algoritmus 2. tartalmazza, feltételezve, hogy az adathalmaz összesen egy ritka esemény csomót tartalmaz.

Az eljárás nem feltétlenül konvergál, a bemeneti paraméterek között tehát a kiugrók becsült aránya mellett egy maximális w lépésszám beállítása is szükséges.

Az NNDB algoritmus mintáját követik az **ALICE** (*Active Learning for Initial Class Exploration*) és **MALICE** (*Modified Active Learning for Initial Class Exploration*) algoritmusok [60], amelyek képesek több ritka osztály egyidejű felderítésére is.

Gyakorlati alkalmazási szempontok

A fentiekben a teljesség igénye nélkül példákat soroltunk fel a ritka események meghatározására és a jó tartománytól történő elválasztására. A felsorolt megközelítések eltérnek abban, hogy milyen szintű előzetes tudást tételeznek fel a vizsgálandó adathalmazról.

Algoritmus 2 NNDB

Input: X: bemenő adatpontok halmaza,
r: ritkák becsült száma
w: a ciklusok maximális száma
Output: *cand*: ritka pont jelölt ($label(cand) = RITKA$)

1. $n = |X|$
2. $K = nr$
3. **for** $i := 1$ **to** n **do**
4. $d_i =$ K. legtávolabbi csomóponttól vett távolság
5. $selected_i =$ **false**
6. **end for**
7. $\varepsilon_1 := min(d_i)$
8. **for** $t := 1$ **to** w **do**
9. $\varepsilon = t \cdot \varepsilon_1$
10. **for** $i := 1$ **to** n **do**
11. $NN(x_i, \varepsilon) = \{x : \|x - x_i\| \leq \varepsilon \}$
12. $n_i = |NN(x_i, \varepsilon)|$
13. **end for**
14. **for** $i := 1$ **to** n **do**
15. **if** $selected_i =$ **false then**
16. $s_i = \max_{x_k \in NN(x_i,\varepsilon)}(n_i - n_k)$
17. **else**
18. $s_i = -\infty$
19. **end if**
20. **end for**
$cand = \arg\max_{x_i} s_i$
21. **if** $label(cand) = RITKA$ **then**
22. **return** *cand* // a $label(cand)$ címkézést egy külső orákulum adja meg;
23. **end if**
24. **end for**

Egyesek előfeltétele az, hogy akár az eloszlásról álljon rendelkezésre használatuk előtt tudás, vagy legalább az eloszlásuk jól jellemezhető legyen.

Ennek megfelelően nincs olyan egyetlen módszer, amely minden környezetben alkalmazható lenne. Szerencsés ugyanakkor, hogy a megkívánt előfeltételeket (például a ritka események csomósodnak) az előzetes vizuális analízissel legalábbis jó közelítéssel ellenőrizni lehet, így mindenképpen célszerű az algoritmikus módszerek használata előtt vizuális feltáró adatelemzés végzése.

Ez utóbbit megkönnyíti az a tény, hogy a vizuális technikákban a ritka események láthatóságát a megjelenítési technikák segítik, ha azok előfordulása független az előfordulási gyakoriságuktól. A két technika kombinációja egészen addig alkalmazható, amikor a szakértő az előzetes analízisben vizuális technikákkal hatékonyan azonosítja a vizsgálandó ritka események, illetve csomóik többségét, majd az ezt követő algoritmikus fázisban már ezen a priori ismeret birtokában alkotja meg a választott módszer a precíz osztályozást.

4. fejezet

Hiányos adatok

4.1. Bevezetés

Az adatelemzés során felmerülő egyik leggyakoribb problémát az jelenti, hogy az adathalmaz nem teljes, egyes értékek hiányoznak. Ez a hiányzás lehet koncepcionálisan *tervezett*, tehát egy adott mintánál (adathalmaz egy sora) a változók (az adathalmaz oszlopai) egy megadott értékkombinációja esetén helyesen nem tartalmaz értéket. Erre példaként egy családi hátteret vizsgáló kérdőív szolgálhat, melyben a nemet, életkort, családi állapotot és a házasságban töltött évek számát kellett megadni. Ekkor az egyedülálló családi állapot esetében nincs értelme a házasságban töltött évek számát vizsgáló kérdésre választ adni, vagyis a hiányzó érték ez esetben szemantikus tartalommal bír.

Minden más esetben a hiányzás *nem tervezett*nek tekinthető. A továbbiakban kizárólag ez utóbbi esetet vizsgáljuk. *Hiányos adatnak* nevezzük tehát a vizsgált adathalmaz azon részeit, melyek különféle (nem tervezett) okokból nem állnak rendelkezésre.

Valós adathalmazok vizsgálata esetén a legtöbb esetben felmerül a hiányos adatok kezelésének kérdése, mivel az adatok egy része jellemzően hiányzik. Leggyakrabban erőforráshiány miatt nincs lehetőség a hiányos értékek újrafelvételére, újramérésére. A problémát az jelenti, hogy az adatelemzési módszerek többsége viszont teljes, azaz hiányos értékek nélküli adathalmazt igényel az elemzés elvégzéséhez.

A hiányzásnak sokféle oka lehet, ezek egy része véletlen, más része lehet akár determinisztikus is. Ez utóbbi egy olyan előre nem látott vagy figyelmen kívül hagyott mechanizmus eredménye, mellyel az adathalmaz kialakításánál nem számoltak (nem keverendő tehát össze a tervezett hiányzással). A hiányzás egy további lehetséges forrása, hogy nem minden esetben áll rendelkezésre a kérdéses információ vagy adott esetben maga a megkérdezett alany tagadhatja meg a válaszadást.

A hiányos adatok nem megfelelő kezelése révén az eredmények jelentősen torzulhatnak, a levont következtetések akár teljesen helytelenek lehetnek. E nem kívánt hatások elkerülése érdekében számos módszert hoztak létre, melyek alkalmazhatósága a hiányzás típusától függ.

4.2. A hiányzás típusai

A hiányos adatok megfelelő kezeléséhez szükséges a hiányzás jellegének ismerete. Minden hiányzó értékkel rendelkező változónál ezt egyedileg meg kell vizsgálni, mivel lehetséges, hogy egy adathalmazon belül akár többféle hiányzástípus is előfordulhat. A hiányzás típusát tekintve az alábbi három osztályt különíthetjük el [87, 56]:

1. *Missing Completely at Random* (MCAR): Teljesen véletlenszerű hiányzás, ahol annak a valószínűsége, hogy egy Y változó értéke hiányzik, nem függ semmilyen megfigyelt $\mathbf{X}$ vagy nem megfigyelhető (rejtett) $\mathbf{Z}$ változó(k)tól.

$$P(Y = m) = P(Y = m|\mathbf{X}, \mathbf{Z}), \tag{4.1}$$

   ahol m a hiányzó értéket jelöli.

2. *Missing at Random* (MAR): Véletlen hiányzás, ahol annak a valószínűsége, hogy Y értéke hiányzik, egy vagy több megfigyelhető változó értékétől függ. Tehát ebben az esetben a hiányzás már valamilyen mértékben jósolható.

$$P(Y = m|X_1 = x_1, \ldots, X_n = x_n) \neq P(Y = m|X_1 = x_1', \ldots, X_n = x_n'), \tag{4.2}$$

   ahol x_1, x_1' és x_n, x_n' rendre X_1 és X_n változók lehetséges értékeit jelölik.

3. *Not Missing at Random* (NMAR): Nem véletlen hiányzás, vagyis Y értéke hiányzásának a valószínűsége nem megfigyelt változók értékeitől függ (akár más megfigyelt változók mellett), azaz

$$P(Y = m|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z},) \neq P(Y = m|\mathbf{X} = \mathbf{x}', \mathbf{Z} = \mathbf{z}'), \tag{4.3}$$

   ahol $\mathbf{x}$, $\mathbf{x}'$ és $\mathbf{z}$, $\mathbf{z}'$ rendre a megfigyelt $\mathbf{X}$ és a nem megfigyelt $\mathbf{Z}$ változók lehetséges értékeit jelölik.

   Továbbá akkor is nem véletlen hiányzásról beszélünk, ha a hiányzás valószínűsége a változó valódi értékétől függ

$$P(Y = m|Y^* = y_1) \neq P(Y = m|Y^* = y_2), \tag{4.4}$$

   ahol Y^* a tökéletes információ esetén rendelkezésre álló, hiányzásmentes Y változót jelöli, y_1 és y_2 pedig a lehetséges értékeit.

   Megjegyezzük, hogy egyes forrásokban a nem véletlen hiányzás ez utóbbi típusát külön osztályba sorolják [56], továbbá helyenként az NMAR elnevezés eltér a szórendet tekintve: *Missing not at Random* (MNAR).

MCAR esetben a hiányzást egy adott valószínűséggel bekövetkező véletlen hibának tekinthetjük, ami nem eredményez torzítást. Ez a legszerencsésebb hiányzási típus, mivel ez kezelhető a legegyszerűbb módon (pl.: teljes minta módszer).

MAR hiányzásnál más, megfigyelt változók értéke alapján becsülhető a kérdéses változó értéke hiányzásának a valószínűsége, továbbá ezt felhasználva következtetni lehet a hiányzó értékre. A legtöbb hiányos adat kezelési módszer ezen a feltevésen alapszik. Az MCAR és MAR hiányzás egy közös tulajdonsága, hogy a hiányzási mechanizmus explicit modellezése elhanyagolható, szemben az NMAR esettel, ahol ez megkerülhetetlen.

Tekintsünk egy példát, ahol egy kérdőív segítségével felmérik a jövedelem és az iskolai végzettség tényezőket. Rendszerint megfigyelhető, hogy a magasabban kvalifikáltak átlagosan magasabb jövedelemmel rendelkeznek, és gyakrabban tagadják meg a válaszadást. Abban az esetben, ha a hiányzás csak a végzettségtől függ, azaz például az egyetemi végzettség esetén a leggyakoribb, de egy-egy kategórián belül nem függ a jövedelem valódi mértékétől, akkor MAR hiányzásról beszélünk. Ha azonban végzettségi kategórián belül igaz az, hogy minél magasabb a jövedelem, annál valószínűbb a hiányzás, akkor már NMAR esetről van szó. Ha általánosan igaz az, hogy a magasabb jövedelműek inkább eltitkolják a jövedelem mértékét, akkor értékfüggő hiányzással kell szembenéznünk. Ilyen esetben a hiányzást modellezni kell, hiszen ha csak elhagynánk a hiányzó értékeket tartalmazó mintákat, akkor a valós populációhoz képest jelentősen torzított adathalmazhoz jutnánk, melyből hiányoznak a legmagasabb jövedelműek.

Összességében elmondható, hogy a hiányzás pontos típusának megállapítása alapvető fontosságú lenne, ám ez már elméletileg sem megvalósítható. Hiszen hogyan bizonyíthatnánk, hogy egy változó értéke nem függ más rejtett változóktól, melyeket definíció szerint nem vizsgáltunk. Bizonyos függőségi mintázatok esetében ugyan ki lehet zárni egy rejtett változó közvetítő hatását két változó közötti függőségi viszonyban, de egy független rejtett változó hatását nem lehet teljesen elvetni [56]. Hasonlóképp, ha a hiányzás csak és kizárólag a hiányos változó valódi értékétől függene, akkor ezt külső referencia nélkül nem tudnánk megállapítani.

Mindezek miatt a gyakorlatban valamilyen feltételezéssel kell élni a hiányzást illetően, ami a leggyakrabban MAR. Ezt célszerű alátámasztani a hiányzások vizsgálatával, például regressziós modellezés segítségével.

Abban az esetben, ha az adathalmaz változóit érintő hiányzás hierarchikus mintázatot alkot, akkor az általános hiányzáskezelő módszerek helyett a mintázat sajátosságait kihasználva hatékonyabb módszert alkalmazhatunk. *Monoton hiányzás*ról akkor beszélünk, ha az adathalmaz változói sorrendezhetőek oly módon, hogy minden X_i, X_j változópárra $i \prec j$ esetén igaz, hogy ha X_j értéke nem hiányos, akkor X_i értéke sem [64]. Tehát ha egy adott változó nem hiányos, akkor minden a sorrendezésben előtte álló változó nem hiányos. Mivel az ilyen jellegű monoton hiányzás –különösen valós problémák esetén– ritka, ezért a továbbiakban kizárólag a nemmonoton hiányzás kezelésével foglalkozunk.

4.3. Hiányos adatok kezelése

A hiányos adatok kezelési módjainak széles skálája ismeretes, az alábbiakban a legelterjedtebb módszereket tekintjük át.

4.3.1. Teljes eset módszer

A legegyszerűbb módszer arra, hogy hiányzásmentes adathalmazt kapjunk az, hogy a hiányos értéket tartalmazó adatsorokat (mintákat) kizárjuk az elemzésből. Ezt teljes eset módszernek (complete cases method) nevezzük, és kizárólag MCAR hiányzás esetén érdemes alkalmazni, mivel csak ebben az esetben nem okoz torzítást [56]. Azonban ekkor is számolni kell azzal, hogy ha a hiányos értékek aránya magas, akkor a szigorú teljességi feltétel miatt az adathalmaz jelentős részét figyelmen kívül kell hagyni. Ez értékes információ elvesztését jelentheti, amely akár az egész elemzést ellehetetlenítheti. Több tíz vagy akár száz faktor vizsgálatánál már különösen megfontolandó, hogy egyetlen változó hiányzó értéke miatt az egész mintát elvessük-e. Különösképpen akkor, ha az adatok forrásául szolgáló mérések erőforrásigényesek és nem pótolhatók. Továbbá, ha eleve relatíve kevés minta áll rendelkezésre, akkor a mintaszám további csökkentése elkerülendő, mivel az a statisztikai vizsgálatok erejét tovább csökkenti.

Ha azonban a hiányzás nem MCAR típusú, akkor a minták elhagyásával torzítjuk a változók értékeinek eloszlását. Ilyen helyzet lép fel minden olyan esetben, amikor a hiányzás magától a hiányos változó valódi értékétől függ, illetve akkor is, ha egy másik megfigyelt változó értékétől függ. Az előbbi esetben a minta kizárásával a hiányos változó eloszlását befolyásoljuk, az utóbbiban pedig a kapcsolódó megfigyelt változó eloszlását. Mindez az elemzés pontatlanságához, továbbá téves konklúziók levonásához vezethet [87].

A teljes eset módszert gyakran helytelenül alkalmazzák egyváltozós elemzéseknél, úgy vizsgálva az adathalmazt, mintha csak a célváltozó és az éppen vizsgált változó hiányzásmentes értékeit tartalmazná (available cases method). Ez azonban azt eredményezi, hogy változónként eltérő a mintaszám, amin az elemzést végzik. Mindez inkonzisztens eredményekhez és helytelen következtetésekhez vezet, ezért a módszer ilyen jellegű alkalmazása mindenképp kerülendő [56].

4.3.2. Ad-hoc módszerek

Az ad-hoc módszerek közé soroljuk azokat az eljárásokat, melyek a változók transzformációján vagy kizárásán alapszanak. Nagyarányú hiányzás ($> 50\%$) esetén az egyik legkézenfekvőbb megoldás az érintett változó(k) kizárása az elemzésből. Azonban ez alapvetően befolyásolhatja az egész vizsgálat kimenetelét. Még egy másodlagos, háttér-információt szolgáltató változó kizárása is értékes információvesztéssel járhat, viszont egy kulcsfontosságú változó elhagyása az elemzés helytelenségét vonhatja maga után. Ez utóbbi esetben tehát érdemes más módszert alkalmazni a hiányzás kezelésére.

A hiányos változó transzformálása alapvetően a hiányzó értékek átkódolását jelenti. Kategorikus változó esetén a hiányzó értéket jelölő új kategóriát lehet létrehozni, folytonos

változónál pedig egy megfelelően választott, máshol nem szereplő értékkel lehet helyettesíteni a hiányzást. Mindez kiegészíthető egy új indikátorváltozó bevezetésével (minden egyes hiányos értékkel rendelkező változóhoz), amely a hiányzás tényét jelöli. Ez a megközelítés (dummy variable adjustment) sokáig népszerű volt, azonban idővel bebizonyították, hogy torzításhoz vezet, ezért alkalmazása nem javasolt [3].

4.3.3. Súlyozás

A súlyozás azt jelenti, hogy a hiányos értékű változó megfigyelt értékeihez súlyokat rendelünk a hiányzási valószínűségeknek megfelelően. Ehhez azonban szükség van a hiányzás mechanizmusát leíró teljes modellre és az ebből származó valószínűségekre. Emiatt csak akkor alkalmazható, ha a hiányzási modell rendelkezésre áll. A súlyozás révén kompenzálhatóak a hiányos értékek, melyeket az adott változó vizsgálatánál a későbbiekben nem vesznek figyelembe. Ezt a módszert szinte kizárólag rétegzett mintavételű szociológiai felméréseknél alkalmazzák, más területeken nem jellemző, de a teljesség kedvéért itt is része a felsorolásnak [64].

4.3.4. Pótlás

A pótlás (imputation) alapú módszerek lényege, hogy a hiányzó értéket a meglévő adat alapján, valamilyen eljárással előállított értékkel helyettesítik. A MAR típusú hiányzás kezelésének ez a preferált formája.

Pótlás alkalmazása során körültekintően kell eljárnunk, tekintettel kell lennünk a hiányzás mértékére, típusára, a változók közötti függésekre, valamint a mintaszámra. Az adathalmaz sajátosságainak megfelelő pótlási módszerrel csökkenthetők a pótlás esetleges negatív következményei, úgymint:

- Nem megfelelő pótlás esetén sérülhetnek az adathoz kapcsolódó modellre jellemző feltevések, például, ha az általunk választott módszer olyan értékkel pótol, ami elvileg nem lehetséges. Ekkor, bár teljes adathalmaz áll elő, a rajta elvégzett elemzés nem lesz érvényes.
- A változónként elkülönítetten végrehajtott pótlás megváltoztathatja az egyes változók közötti függéseket.
- A változók varianciáját nagymértékben alulbecsülhetjük, ha a pótolt értékeket megfigyeltnek tekintjük.

A továbbiakban a hiányzó adatok pótlására használt módszercsaládok ismertetésére kerül sor.

Heurisztikus pótlás

E módszercsoportba olyan egyszerű eljárások tartoznak, melyek egy adott eljárás szerint pótolnak egy a tárgyterülettől független heurisztika által nyert rögzített vagy véletlen értékkel. A legáltalánosabb formái közé az alábbiak tartoznak:

- *Egyszerű véletlen pótlás*: a hiányos értékekkel rendelkező változó adathalmazban szereplő értékeivel történik a pótlás véletlenszerűen. Minden egyes hiányzó értéknél sorsolunk, figyelembe véve az értékek eloszlását. A módszer előnye, hogy egyszerű és gyorsan megvalósítható, ugyanakkor figyelmen kívül hagyja a változók közötti függőségeket, és nem használja fel az adathalmazban jelen lévő többlet információt a pótláshoz.

- *Random hot deck*: az adott hiányos adatsorhoz rögzített szempontok szerint leghasonlóbb adatsorból származtatja a pótlásra felhasznált értéket. Ezt a módszert elsősorban kérdőíves felméréseknél használták, ahol hiányzás esetén néhány választott tényező alapján (pl.: nem, életkor) választottak a hiányos adatsorhoz hasonlót az adathalmazból, és az ott szereplő értékkel pótolták. A „*hot deck*" arra utal, hogy a vizsgált adathalmaz alapján történt a pótlás, szemben a „*cold deck*" módszerrel, ahol egy korábbi, hasonló felmérés adathalmazából származott a pótolt érték [56]. Manapság a pótlásnak ezt a formáját nem használják, ehelyett az ezzel rokon legközelebbi szomszéd módszerek használatosak.

- *Legközelebbi szomszéd pótlás*: a hiányos értéket tartalmazó adatsorhoz leginkább hasonló adatsorokban szereplő értékkel történő pótlást jelent. Tehát egy W változó esetében hiányos i-edik adatsornál ($d^{(i)}$) megvizsgálja az ahhoz valamilyen $\mathcal{S}$ metrika szerint hasonló adatsorokat $d^{(j_1)} \dots d^{(j_k)}$, melyekben W nem hiányos, és ezen $W^{(j_1)} = w_{j_1} \dots W^{(j_k)} = w_{j_k}$ értékek közül a legvalószínűbb értékkel pótolja $W^{(i)}$-t. E módszernek számos különböző változata lehetséges, például csak a leghasonlóbb adatsort figyelembe vevő pótlás, vagy a k legközelebb eső adatsort (szomszédot) figyelembe vevő pótlás [2]. Egy további lehetőség, hogy W változó mellett $X_1, \dots X_k$ változók legvalószínűbb együttes értékkonfigurációja alapján történik a pótlás.

- *Átlag pótlás*: értelemszerűen a meglévő értékből számított átlaggal történő pótlást jelenti. Akár a teljes adathalmaz alapján, akár a célváltozó szerinti adott kategórián belül számítjuk, jelentős torzítást eredményez, ha egy rögzített értékkel történik a pótlás. Ennek eredménye, hogy mesterségesen csökken a pótolt változó varianciája, illetve a rendelkezésre álló mintaszám úgy nő (a kipótolt adatsorokkal), hogy új információ nem kerül az adathalmazba a pótolt változó szempontjából, tehát tévesen felülbecsüljük a mintaszámot. Továbbá az értékek eloszlása jelentősen torzul, különösen nagy hiányzási arány esetén [80]. Mindezek miatt az átlaggal vagy más rögzített értékkel (pl. mediánnal) történő pótlás nem javallott [87, 56].

Regressziós pótlás

Egy adott Y változó értékének regressziós pótlása más megfigyelt $X_1, \dots, X_r$ változók alapján számított regressziós modell alapján történik. Folytonos változó esetén lineáris regressziót, kategorikus változó esetén pedig logisztikus regressziót célszerű alkalmazni. Jelentős különbség a korábbi módszerekhez képest az, hogy ez a módszer a pótlandó hiányos változó függőségeit figyelembe veszi (az Y-nal függőségi kapcsolatban álló X_i változók

szerepelnek a modellben nem nulla együtthatóval). Hátránya az, hogy ilyen formában ez is egy determinisztikus pótlásnak tekinthető, ily módon a variancia torzulása itt is fellép, ahogy az átlagpótlásnál [65]. Ennek kiküszöbölésére egy véletlenszerű hibataggal kell bővíteni a regressziós egyenletet. Lineáris regresszió esetén ez a következő:

$$Y = \beta_1 \cdot X_1 + \ldots + \beta_r \cdot X_r + \epsilon, \tag{4.5}$$

ahol ϵ jelöli a predikciós hibát megtestesítő hibatagot.

A regressziós pótlásnak számos változata létezik, a különbség származhat abból, hogy csak teljes adatsorokat vesz figyelembe, vagy iteratív jelleggel a már pótoltakat is, továbbá az egyes változók között van-e meghatározott pótlási sorrend, és így a már teljesen pótolt változókat használja a még hiányosak pótlásához (iteratív regressziós pótlás). Egy további fontos szempont az egyes változók pótlásának kölcsönhatása, ugyanis, ha egymástól teljesen függetlenül végzünk változónként regressziós számításokat, akkor összességében inkonzisztens adathalmazt kaphatunk, melyben jelentősen torzulnak a változók függései [56]. Erre az egyik lehetséges megoldás több változó együttes pótlása egy erre alkalmas többváltozós pótlási modell alapján. Ugyanakkor egy ilyen modell kialakítása, különösen relatíve sok változó esetén, összetett, számításigényes feladat.

Pótlás háttértudással

A hiányos értékek pótlását jelentősen elősegítheti az esetleg rendelkezésre álló tárgyterületi háttértudás. Emiatt olyan vizsgálatok folyamán, melyeknek célja az exploráció, és a vizsgált faktorok közötti függőségek ismeretlenek, háttértudásra építő metódusok nem vagy csak igen korlátozottan alkalmazhatóak. A korábbiakban ismertetett módszerek egy része kibővíthető oly módon, hogy a háttértudás felhasználásával hatékonyabban működjön. Többek közt, ha a változók egy halmazának jellemző értékkonfigurációi vagy az értékek közötti függőségek ismertek, akkor ez felhasználható a pótlásnál egyes értékek kizárására. Így szűkíthető adott esetben a pótlásnál használandó értékek halmaza, ami az összes lehetséges értékkombinációhoz képest jelentősen csökkentheti a számítási igényt. Hasonlóképp, az adatsorok közti hasonlóság vizsgálatánál figyelembe veendő változók száma a háttértudás alapján pontosabban meghatározható. Mindez a legközelebbi szomszéd típusú pótlási módszereknél jelenthet előnyt.

Likelihood alapú megközelítés

A likelihood alapú módszerek alapvető célja az Y célváltozó $p(Y|X, \theta)$ feltételes valószínűségi eloszlását leíró θ paraméterek meghatározása maximum likelihood becsléssel. Ennek alapjául az *EM* (expectation-maximization) algoritmus szolgál, ami egy kétlépcsős, determinisztikus, iteratív algoritmus [33]. A becslési lépés során (expectation step) minden hiányos értéknél több lehetséges értékkel kell elvégezni a pótlást, és minden egyes értéknél meg kell becsülni, mennyire valószínű az adott érték a többi megfigyelt érték függvényében. Az ezt követő maximalizálási lépésben (maximization step) az együttes valószínűségeloszlás várható log-likelihoodjának maximalizálására kerül sor az előző lépésben előállított

értékek felhasználásával. Az ennek következtében adódó legjobb értékek adják a következő becslési lépés alapját. A több iterációt követően eredményként előálló paraméterértékek a maximum likelihood becsléshez konvergálnak, a lokális maximumokba való beragadás lehetőségével azonban számolni kell.

Bayes-háló alapú megvalósítás esetén az alábbiak szerint írható fel az együttes valószínűség-eloszlás likelihoodja. Jelölje G a Bayes-háló struktúráját leíró aciklikus irányított gráfot, mely az $\mathbf{X}$ változók közötti függőségeket reprezentálja. Az együttes valószínűség-eloszlás a struktúra által meghatározott módon faktorizálható:

$$P(\mathbf{X}|G,\theta) = \prod_{i=1}^{k} P(X_i|PA_i,\theta) = \prod_{i=1}^{k} \theta_{X_i|PA_i}, \tag{4.6}$$

ahol PA_i az X_i változó szülői halmaza G-ben. Egy adott $\theta_{X_i|PA_i}$ paraméterezésnél figyelembe kell venni az X_i változó lehetséges értékeit (x_i) és a szülők lehetséges értékeinek konfigurációit (pa_i). Mindez szükséges a $\mathcal{L}(\theta|G,D)$ likelihood számításához, mivel ahhoz az összes lehetséges paraméterezés valószínűségét ki kell számítani a rendelkezésre álló D adathalmaz alapján:

$$\mathcal{L}(\theta|G,D) = \prod_{i=1}^{k} \prod_{x_i} \prod_{pa_i} \theta_{x_i|pa_i}^{\#(x_i,pa_i)}, \tag{4.7}$$

ahol $\#(x_i,pa_i)$ egy adott érték-konfigurációhoz tartozó minták számát jelöli D-ben. Ez a szorzatforma azonban csak teljes adatra áll fenn, ha a hiányos értékeket nem pótolnánk, akkor mintánként minden hiányos értékre ki kellene „átlagolni”, azaz az összes lehetséges pótlással számolni és azok átlagát venni [92]. Mindez jelentősen megbonyolítaná a likelihood számítását, az EM algoritmus alkalmazása erre nyújt alternatív megoldást.

Az EM algoritmust felhasználó Bayes-háló alapú módszereknek többféle változata létezik, úgymint a strukturális EM [46] és a bayesi strukturális EM [47]. A strukturális EM Bayes-háló struktúrák tanulását teszi lehetővé komplexitást büntető likelihood score-ok alapján, míg a bayesi strukturális EM egy közvetlenül számolt Bayes-score alapján teszi ugyanezt.

Bayesi módszerek

Bár számos Bayes-háló alapú pótlási módszer létezik, ezeknek csak egy része tekinthető teljesen bayesi módszernek. Az előbbiekben tárgyalt likelihood alapú módszerektől szemléletben elkülönülnek, holott azok jelentős része szintén Bayes-háló alapú megvalósítással rendelkezik. Bayesi megközelítésben a cél a $P(\theta|G,D)$ *a posteriori* paramétereloszlás meghatározása, melyhez szükséges egy $P(\theta|G)$ *a priori* eloszlást definiálni a paramétertér felett. A Bayes-tétel alapján az alábbi összefüggés adódik e mennyiségek között:

$$P(\theta|G,D) \propto \mathcal{L}(\theta|G,D) \cdot P(\theta|G), \tag{4.8}$$

tehát a $P(\theta|G,D)$ a posteriori valószínűség arányos a likelihood és az a priori valószínűség szorzatával.

E módszerek közé tartozik a Bayes-hálók strukturális tulajdonságait felhasználó, Markov-takaró alapú pótlás, melynek lényege, hogy minden változónál csak a hozzá képest releváns változók értékeit veszi figyelembe a lehetséges pótlási értékek meghatározásánál [92]. Egy további ide sorolható módszer a sztochasztikus szimuláció alapú paraméterbecslést végző *data augmentation* algoritmus [108], illetve ennek kapcsán meg kell említeni egy ehhez hasonló, de más szemléletű módszert: a *bound and collapse* algoritmust [100].

Összességében e pótlási módszerek összetettségükből fakadóan kevésbé elterjedtek a korábbiakban bemutatott módszerekhez képest, ugyanakkor a többváltozós elemzés alapjait jelentő változók közötti függőségi kapcsolatokat ezek torzítják a legkevésbé.

4.3.5. Többszörös pótlás

A többszörös pótlás (multiple imputation) voltaképp egy általánosan alkalmazható paradigma a legtöbb nemdeterminisztikus pótlási módszer esetében. Lényege, hogy a hiányzás okozta bizonytalanság modellezésére explicite sor kerül azáltal, hogy egy helyett több kipótolt adathalmazt állítunk elő. A többszörös pótlás általánosan három lépésből áll:

1. A hiányos adatok pótlására egy adott módszerrel előállítjuk a lehetséges értékeket és végrehajtjuk a pótlást.

2. A keletkező teljes adathalmazokon elvégezzük a statisztikai elemzést, következtetést (ezek teljes adathalmazt igényelnek).

3. A végeredményeket összesítjük, figyelembe véve a pótolt értékek bizonytalanságát, azaz kiszámítjuk az eredmények átlagos értékét, konfidencia-intervallumát, illetve varianciáját.

Tegyük fel, hogy regressziós pótlás esetén alkalmaztunk többszörös pótlást, ekkor a β regressziós koefficiensek meghatározása az egyik cél. Ilyen esetben m végrehajtott pótlást követően egy β_i koefficiens átlagos értéke ($\hat{\beta}_i$) és átlagos szórásnégyzete (S_i) az alábbi formában áll elő:

$$\hat{\beta}_i = \frac{1}{m}\sum_{m=1}^{M}\beta_i^{(m)}, \quad S_i = \frac{1}{m}\sum_{m=1}^{M}(s_i^2)^{(m)}, \tag{4.9}$$

ahol $\beta_i^{(m)}$ az adathalmaz m-edik pótlásánál előálló értéke β_i koefficiensnek, $(s_i^2)^{(m)}$ pedig a hozzátartozó szórásnégyzet. A teljes variancia Var_β a pótlásokon belüli (S_i) szórásnégyzet mellett a pótlások közötti szórásnégyzet (U_i) figyelembe vételével adódik [56]:

$$Var_\beta = S_i + (1 + \frac{1}{m}) \cdot U_i, \quad \text{ahol} \tag{4.10}$$
$$U_i = \frac{1}{m-1}\sum_{m=1}^{M}(\beta_i^{(m)} - \hat{\beta}_i)^2$$

A gyakorlatban legtöbbször regresszió alapú pótlásnál alkalmazzák a többszörös pótlás elvét, azonban léteznek alternatív többszörös pótlást megvalósító módszerek, úgymint a *chained equations* módszer [116]. Ez az algoritmus minden változónál egyedi pótlási modell alapján generál értéket. Iterációnként egy hiányzó értéket pótol egy adott változónál, ami a következő változó pótlásakor már számít a pótolt érték meghatározásánál.

5. fejezet

Vizuális analízis

Az adatvizualizáció, akárcsak a matematikai statisztika, napjaink természet- és társadalomtudományainak és mérnöki gyakorlatának alapvető eszköze. Fejlődése összefonódik alkalmazási területeinek fejlődésével; már a 10. századból ismert olyan ábra, mely hét égitest elhelyezkedésének változását demonstrálja térben és időben – mai szóhasználatunkban idősorként [52]. A 19. század első felére a (statikus) statisztikai grafika alapvető eszközeinek többsége kialakult: így például az oszlopdiagram (*bar chart*[1]), a hisztogram, az idősorábra (*time series plot*), a szintvonal-ábra (*contour plot*) vagy a szórásdiagram (*scatterplot*) [51]. A 70-es évektől kezdve az adatvizualizáció mai gyakorlatához vezető fejlődési folyamatot alapvetően befolyásolta a felderítő adatanalízis (*exploratory data analysis*), mint önálló statisztikai diszciplína kialakulása és a számítógéppel megvalósított adatábrázolás lehetővé, majd gyakorlatilag egyeduralkodóvá válása. Bár az egyes szakterületek szükségletei még ma is életre hívnak újabb és újabb diagram-típusokat (melyek sokszor csak az ismert típusok variációi), a modern statisztikai adatvizualizáció legfontosabb minőségi újításai a magas dimenziójú statisztikai adatok kezelése, valamint az interaktív és dinamikus megjelenítési technikák. Kialakulóban vannak, de messze nem kiforrottak azok az általános vizualizációs módszerek, melyek segítségével extrém méretű adatkészletek – divatos szóhasználatban: „Big Data" problémák – is célszerűen szemléltethetővé válnak (lásd pl. [120]).

Az adatvizualizáció és a segítségével megvalósított vizuális analízis rendkívül szerteágazó témák; fejezetünk célja a sokváltozós statisztikához kapcsolódó alapvető és általános vizualizációs technikák és az ezekre épülő elemzési megközelítések ismertetése. Alkalmazási útmutatóként – különösen a statikus diagramok tekintetében – az `R` [91] nyílt forráskódú és ingyenes statisztikai számítási környezetben rendelkezésre álló megoldásokra fogunk hivatkozni; meg kell azonban említenünk, hogy a diagramok többségét ma már mindegyik meghatározó adatelemzési eszköz támogatja.

[1] A jellemzően angol nyelvű, nemzetközileg elfogadott terminológia helyett a fejezetben törekszünk a magyar megfelelők használatára – az eredeti megjelölésével. Felhívjuk azonban az olvasó figyelmét arra, hogy a terület számos szakkifejezésének nincs egyértelműen és általánosan elfogadott magyar fordítása.

5.1. Felderítés, megerősítés és szemléltetés

Bár az adatvizualizáció szinte minden adatelemzési feladatban megjelenik, súlya és főként alkalmazásának módja az adatelemzés célja és fázisa szerint más és más. A vizualizáció eszköze lehet *a*) az elemzendő adatok *megértésének* és *hipotézisek megsejtésének*; *b*) hipotézisek és modellek megerősítésének, vagy *c*) az eredmények – lehetőleg minél szemléletesebb – *prezentálásának*. Az első két tevékenység-kategóriára szokásosan mint *felderítő* adatelemzés (*Exploratory Data Analysis – EDA*), illetve *megerősítő* adatelemzés (*Confirmatory Data Analysis – CDA*) hivatkozunk.

A John W. Tukey amerikai matematikus és statisztikus munkássága által megalapozott EDA leginkább egy „statisztikai tradíció", melyet röviden a következőképp jellemezhetünk [9].

1. Az adatelemzési folyamat nagy hangsúlyt fektet az adatok általánosságban „megértésére" - az adatok által leírt rendszerről, az adatokon belüli és az adatok és általuk leírt rendszer közötti összefüggésekről minél teljesebb fogalmi kép kialakítására.

2. Az adatok grafikus reprezentációja, mint ennek eszköze, kiemelt szerepet kap.

3. A modelljelölt-építés és a hipotézisek felállítása iteratív módon történik, a modell-specifikáció – reziduum-analízis – modell-újraspecifikáció lépéssorozat ismétlésével.

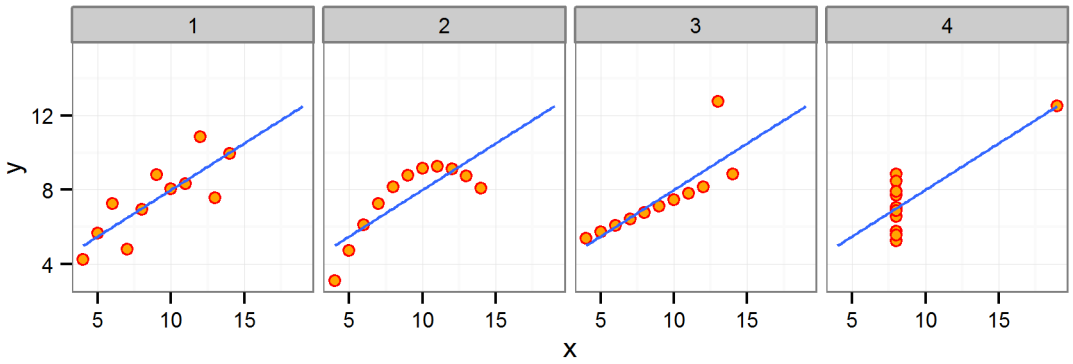
4. A CDA-val összevetve előnyt élveznek a robusztus statisztikai mértékek és a részhalmazok analízise.

5. Az EDA „detektívmunka" jellege miatt nem szabad bevett módszerekhez dogmatikusan ragaszkodni; a folyamatot köztes sejtéseink mentén flexibilisen kell alakítani, előfeltételezéseinket (pl. hogy két változó között korreláció „szokott" fennállni) megfelelő szkepszissel kezelve.

Természetéből adódóan az EDA legtöbbször egy erősen ad-hoc folyamat; jellemezhető úgy is, mint az adatok (jellemzően) alacsony dimenziószámú grafikus vetületeinek intuíció- és szakterület-specifikus tudás által vezérelt bejárása addig, amíg klasszikus statisztikai elemzést érdemlő hipotézisekre nem jutunk.

A megfelelő vizualizáción keresztül összefüggések megsejtésének iskolapéldája Dr. John Snow története. Dr. Snow 1854-ben egy londoni kolerajárvány alkalmával egy pontozott térképet használva a halálesetek vizualizálására felfedezte, hogy a járvány oka egy fertőzött kút a Broad Streeten; a kút nyelét leszereltetve a járvány megszűnt. (A történet második része valószínűleg nem igaz; lásd [86].) A CDA túl korai, EDA-t nélkülöző alkalmazásának veszélyeire a klasszikus intő példa pedig „Anscombe négyese" (*Anscombe's quartet*)[5]: négy kétváltozós adatkészlet, melyeknek ugyan átlaga, varianciája, korrelációja és regressziós egyenese – azaz klasszikus statisztikai jellemzői – megegyeznek, mégis minőségileg különböző összefüggéseket rejtenek (5.1. ábra)[2].

[2]Anscombe négyese az `R` beépített, `anscombe` néven előhívható adatkészlete.



5.1. ábra. Anscombe négyese szórásdiagramokkal vizualizálva

Az alfejezet további részében bemutatjuk a vizuális EDA során leggyakrabban alkalmazott diagram-típusokat. Bár ezeket sokszor alkalmazzuk a CDA támogatására is, ott jellemzően modellspecifikus vizualizációra van szükség (pl. lineáris regresszió reziduumainak vizsgálata).

5.2. Egydimenziós diagramok

Egy sokváltozós megfigyeléskészlet változóinak peremeloszlás-vizsgálata az EDA első lépései között szokott szerepelni. Amennyiben a változó kategorikus, úgy a közismert oszlopdiagram és változatai szolgálhatnak a kategóriák számosságának vizualizációjára.

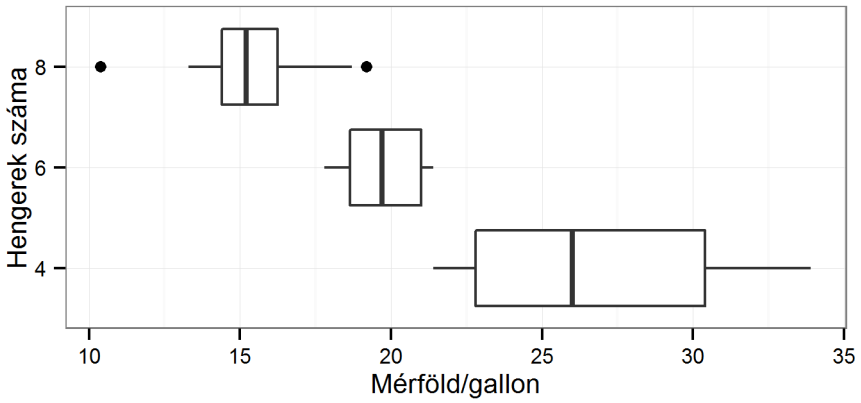
Az egy, folytonos változóra vonatkozó megfigyeléseket reprezentálhatjuk közvetlenül egy tengelyen, ún. pontdiagramon (*dot plot*) - a szórásdiagram egyfajta „leskálázásaként". Az egydimenziós, folytonos esetben azonban jellemzően nem közvetlenül a megfigyeléseket, hanem vagy meghatározó leíró statisztikáikat, vagy eloszlásukat szeretnénk vizualizálni. Mindkét esetben jellemző – legalább a felderítő analízis során – hogy *nemparaméteres* technikák alkalmazására törekszünk, azaz a változó eloszlásával kapcsolatban nem kívánunk előfeltételezésekkel élni.

5.2.1. A doboz-ábra

Egy egyváltozós megfigyelés-sokaság alapvető leíró statisztikáinak legnagyobb kifejezőerejű vizualizációs eszköze a doboz-ábra (*box plot*) és különböző variációi [85]. A doboz-ábra öt alapvető jellemzőt szemléltet: a minimumot, az első kvartilist, a mediánt, a harmadik kvartilist[3] és a maximumot. A kvartilisek távolságát, mint az eloszlás „középső felét", egy

[3]Az első és a harmadik kvartilis helyett precízebb lenne alsó és felső „sarkalatos pontról" – *hinge* – beszélnünk. Az alsó sarokpont definíció szerint az $([(n+1)/2]+1)/2$-ed rendű statisztika, mely ha nem egész, akkor a szomszédos statisztikák átlagát használjuk; a gyakorlatban ez azonban jó közelítéssel az első kvartilis. A felső sarokpont hasonlóan definiálható.

„doboz” reprezentálja; az első kvartilis alatti és a harmadik feletti részt jellemzően egy vonal, „bajusz” (*whisker*). Egy jellemző variáció, hogy ez a vonal nem a minimumig és a maximumig nyúlik, hanem legfeljebb a kvartilisek közötti távolság (*Inter-Quartile Range*, $IQR = Q_3 - Q_1$) 1,5-szereséig — az ezen kívül eső megfigyeléseket pontként ábrázolva. Emellett – bár a doboz-ábrát egydimenziós vizualizációs technikaként a legegyszerűbb bevezetni – a gyakorlatban legtöbbször a megfigyeléseket egy kategorikus változó mentén részhalmazokra bontva használjuk. Az 5.2. ábra példa doboz-diagramja is kétdimenziós ebben az értelemben[4].



5.2. ábra. Példa doboz-ábrára: személygépjárművek üzemanyag-hatékonysága

Egy doboz-diagram önmagában egy egyváltozós adatkészlet centrális tendenciája, diszperziója és ferdesége (*skewness*) felmérésének praktikus eszköze. Több eloszlás esetén (pl. kategorikus változó mentén bontás mellett) ezen jellemzők gyors, vizuális összehasonlítására is alkalmas. Hátránya, hogy az eloszlást nagyon erősen absztrahálja; így például a multimodalitás jellemzően nem olvasható le róla.

5.2.2. Hisztogram

Legyen $x \in \mathbb{R}$ változó, melyen az $a, b \in \mathbb{R}$ intervallumon n megfigyeléssel rendelkezünk. Képezzük $[a, b)$ egy L nemátfedő intervallumokból (*bin*-ekből, cellákból) álló partícionálását: $T_l = [t_{n,l}, t_{n,l+1}), l = 0, 1, 2, \ldots, L-1$, ahol $a = t_{n,0} < t_{n,1} < t_{n,2} < \ldots < t_{n,L} = b$. Legyen I_{T_l} az l-ik cella indikátor-függvénye és legyen $N_l = \sum_{i=1}^{n} I_{T_l}(x_i)$ a T_l-be eső minták száma ($l = 0, 1, 2, \ldots, L-1$, $\sum_{l=0}^{L-1} N_l = n$). Ekkor a hisztogram, mint a változó eloszlásának becslője, a következőképp definiálható ([69], 80. oldal):

$$\hat{p}(x) = \sum_{l=0}^{L-1} \frac{N_l/n}{t_{n,l+1} - t_{n,l}} I_{T_l}(x).$$

[4] Az ábra az `R` beépített `mtcars` adatkészlete felett készült, mely 32, 1973–74-es személygépkocsi-modell tíz aspektusát írja le.

A gyakorlatban általában azonos, de a minták száma által befolyásolt cellaszélességet használunk ($h_n = t_{n,l+1} - t_{n,l}, l = 0, 1, 2, \ldots, L-1$). A struktúra vizualizációja közismert; az 5.7. ábrán láthatunk két példát. A hisztogram az EDA szempontjából legfontosabb hátránya, hogy alakja erősen érzékeny mind az első cella kezdőpontja, mind pedig a cellaszélesség megválasztására. Becslőként is komoly hiányosságai vannak; numerikus sűrűségbecslésre legtöbbször alkalmasabbak a kernel-sűrűségbecslők (lásd pl. [69], 4.5. fejezet). Vizuális elemzés során azonban egyszerűbb, az „adatokhoz közelebb eső" interpretálhatóságuk miatt mégis a hisztogramok előnyben részesítése javasolt.

5.3. Kétdimenziós diagramok

Statikus vizualizáció esetén a két változó közötti interakciót szemléltető, általános célú diagramok szempontjából a két kategorikus változó esete érdemel kiemelt figyelmet. A folytonos-folytonos esetben alkalmazható szórásdiagramok és hőtérképek (*heat map* – valójában 2D hisztogramok) közismertek; a kategorikus-folytonos esetben alkalmazhatunk pl. kategóriánként alkalmasan színezett hisztogramokat (lásd pl. később az 5.8. ábrán) vagy a már bemutatott kondicionált doboz-diagramokat.

A kétváltozós kategorikus-kategorikus esetben elsődleges célunk a kategória-kombinációk relatív számosságának felmérése lehet. Erre általában a mozaik diagram (*mosaic plot*) vagy a fluktuációs diagram (*fluctuation plot*) a legalkalmasabbak. Ezek azonban n-dimenziós diagramtípusok, melyeknek a két változó megjelenítése speciális esete.

5.4. n-dimenziós diagramok

Kettőnél több változó megjelenítésére két alapvető diagramtípust mutatunk be: a tisztán kategorikus esetre a mozaik diagramot (és a variánsának tekinthető fluktuációs diagramot), a tisztán folytonos esetre pedig a párhuzamos koordináta (*parallel coordinates*) diagramot. A párhuzamos koordináta diagramon kategorikus változók is megjeleníthetőek a kategóriákhoz számérték rendelésével, azonban mint látni fogjuk, ez jellemzően csak akkor eredményezhet „jól olvasható" diagramot, ha a kategorikus változónak megfelelő tengelyek nem szomszédosak.

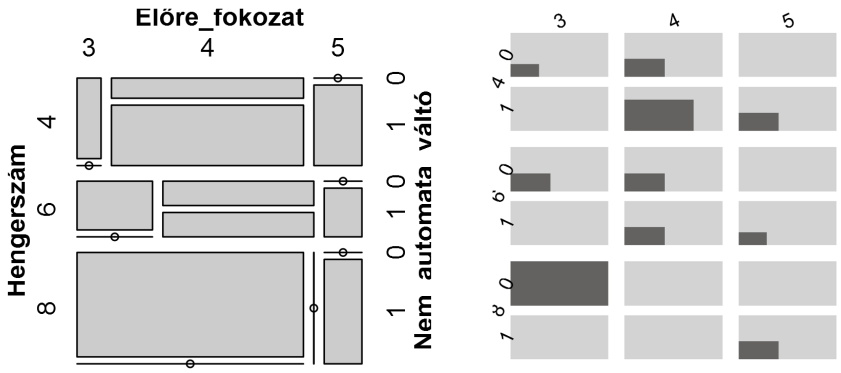
Megjegyezzük, hogy a folytonos esetben elterjedten használt még a szórásdiagram-mátrix (*scatterplot matrix*, *SPLOM*); hogy egy sokváltozós adatkészlet változó-vektorát megfeleltetjük egy mátrix sorainak és oszlopainak, a cellákban pedig a megfelelő változópárok szórásdiagramját helyezzük el. A SPLOM és variánsai – a párhuzamos koordinátákkal ellentétben – magasabb változószámnál korlátozott használhatósága miatt mélyebb bemutatásuktól eltekintünk.

5.4.1. Mozaik és fluktuációs diagram

A mozaik diagram kategorikus változók érték-kombinációi előfordulási gyakoriságainak területarányos vizualizációja. Az ábrát alkotó „csempéket" vagy „lapokat" (*tiles*) egy négy-

zet rekurzív vízszintes és függőleges darabolásával kapjuk. Az 5.3. ábrán látható példa az `mtcars` adatkészlet három változóját helyezi el mozaik diagramon, hengerek száma, előre fokozatok száma és a váltó nem automata volta sorrendben. Egy negyedik változó az előre fokozatok száma alá eső, annak értékeit rendre saját lehetséges értékeivel aláosztó faktorként jelenne meg. Egy ötödik változó a hengerszám – nem automata váltó bontást finomítaná tovább, és így tovább. A mozaik diagramok effektív olvashatósága természetesen adatfüggő is, de elmondható, hogy körülbelül 8 változónál többet általában semmiképp sem érdemes alkalmazni. Mindemellett az olvashatósággal már 4–5 változó esetén adódhatnak problémák.

A fluktuációs diagram a mozaik diagram magasabb dimenziószámnál nehezen olvashatóságát próbálja orvosolni. Az egyes érték-kombinációkhoz azonos méretű lapokat rendelünk, ezáltal a kombinációk könnyen beazonosíthatóvá válnak. A legnagyobb elemszámú lapot teljesen kitöltjük, a többit pedig az előfordulások relatív gyakorisága alapján területarányosan. (A kitöltő téglalapok oldalaránya a mozaik diagramok rekurzív bontásával szemben egységesen ugyanaz; lásd 5.3. ábra.) Hátránya, hogy a kitöltő idom nem hordozza közvetlenül az egy-egy változóban értelmezett relatív gyakoriságot vizuális információként.

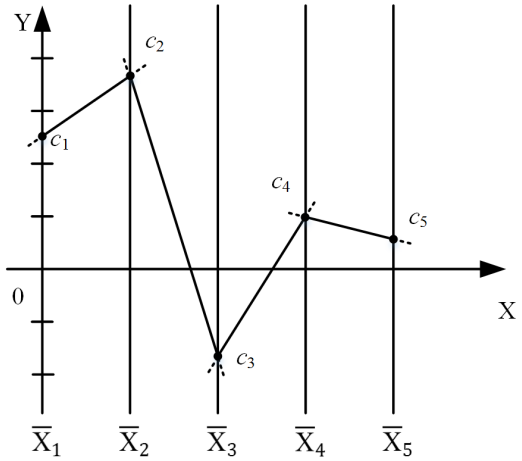


5.3. ábra. Mozaik diagram és fluktuációs diagram

5.4.2. A párhuzamos koordináta ábra

A párhuzamos koordináta diagram [68], mint a sokváltozós relációk vizualizációjának eszköze, egy igen egyszerű ötleten alapul. A klasszikus Descartes-féle ortonormált koordinátarendszer „gyorsan kimeríti a síkot"; már három dimenzió esetén is projekciókra van szükségünk. Ennek oka az, hogy a tengelyek merőlegesek (de legalábbis szöget zárnak be). A párhuzamos koordináták ezzel szemben a szokásos Descartes-féle koordinátákkal ellátott $\mathbb{R}^2$ euklideszi síkban egymástól azonos (egységnyi) távolságra elhelyezi az $\mathbb{R}$ valós egyenes N másolatát, és ezeket használja tengelyként az $\mathbb{R}^N$ N-dimenziós euklideszi tér pontjainak ábrázolására. A $(c_1, c_2, \ldots, c_n)$ koordinátákkal rendelkező $C \in \mathbb{R}^N$ pont képe az a teljes poligon-vonal, melynek N darab, a szegmenseket meghatározó pontjai rendre a

párhuzamos tengelyekre eső $(i-1, c_i)$ pontok $(i = 1, \ldots, N)$. Ezt a kölcsönösen egyértelmű leképezést szemlélteti az 5.4. ábra[5].



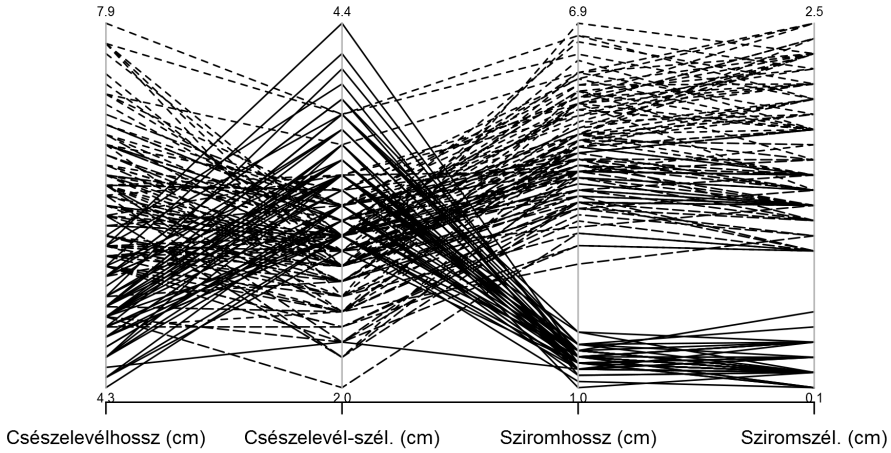
5.4. ábra. $\mathbb{R}^N$-beli pont ábrázolása párhuzamos koordinátákkal

Figyeljük meg, hogy a módszer „helyigénye" lineárisan skálázódik a változók számában, szemben pl. a szórásdiagram mátrix négyzetes növekedésével. Így a párhuzamos koordináta diagram magas dimenziószámú adatkészletek ponthalmazainak projekció nélküli *áttekintésére* különösen alkalmas eszköz. Az 5.5. ábrán a híres Fisher-féle írisz-adatkészlet[6] párhuzamos koordinátákra leképezése látható. Példánk egyben azt is jól szemlélteti, hogy miért kisebb a párhuzamos koordináták jelentősége a statikus vizualizáció területén, mint az interaktív technikáknál: nagyszámú pont esetén az ábra gyorsan átláthatatlanná válik, különösen faktorváltozók szerinti megkülönböztető színezés nélkül. Amennyiben azonban rendelkezésre állnak a megfelelő interakciók – mint pl. részhalmaz kiválasztása, tengelyen intervallum kiválasztása –, a párhuzamos koordináták különösen hatékony EDA eszközt adnak kezünkbe.

Ennek fő oka, hogy számos síkbeli és térbeli alakzat a párhuzamos koordinátákkal ábrázolás során jellegzetes mintává képződik le, így az EDA felfogható egyfajta mintafelismerési problémaként. A talán legegyszerűbb eset ennek szemléltetésére a pont-vonal dualitás. Mint láttuk, egy pont két, szomszédos párhuzamos koordináta-tengelyen értelmezett képe egy szakasz. Az ezen két tengelyhez tartozó koordináták síkjában felvett egyenes képe viszont párhuzamos koordinátákban egy *pont* abban az értelemben, hogy az egyenes pontjai leképezésének tekinthető szakaszok egy pontban fogják metszeni egymást. (Nem feltétlenül a két párhuzamos tengely között.) Ez azt jelenti, hogy amennyiben egy két tengely közötti szakaszsereg egy pontban metszi egymást egy párhuzamos koordináta ábrán, úgy a tengelyek (Descartes) koordinátáinak síkjában egy egyenesre illeszkednek – ez nem más, mint a lineáris korreláció „képe" párhuzamos koordinátákban. Ezt szemlélteti az 5.6. ábra. Figyeljük meg, hogy a párhuzamos egyenesek egymáshoz képest függőlegesen

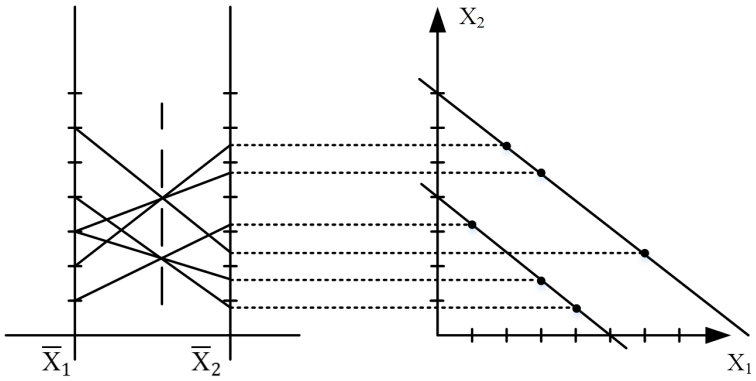
[5] Az ábra forrása: [68], 3. oldal

[6] `iris` néven beépített adatkészlet az R-ben.



5.5. ábra. Az írisz-adatkészlet párhuzamos koordinátákban, fajok szerinti vonaltípusokkal

eltolt pontokként jelennek meg! Könnyű belátni, hogy ehhez hasonlóan egy párhuzamos tengelyek közötti pont vízszintes eltolása pedig a megfelelő egyenes „forgatásának” felel meg.



5.6. ábra. Azonos egyenesre eső pontok leképezése párhuzamos koordinátákra

Az ismert, egyéb alakzatokra vonatkozó, illetve magasabb dimenziószámú minták (mint pl. a hiperbola $\leftrightarrow$ ellipszis leképezés, $\mathbb{R}^3$-ban azonos síkban elhelyezkedés vagy N-dimenziós egyenes – $N-1$ nem párhuzamos hipersík metszete – felismerése) bemutatására itt nincs lehetőségünk; az olvasó figyelmébe ajánljuk a leginkább autoritatívnak tekinthető összefoglaló művet [68].

5.4.3. Eszköztámogatás

Az 5.1. táblázat megadja a bevezetett, illetve hivatkozott diagramtípusokat megvalósító `R` függvényeket. Az `R`-ben különösen a grafika területén igaz, hogy ugyanannak a funkciónak több, egymástól képességekben és kifinomultságban különböző megvalósítása is elérhető. Törekedtünk a legegyszerűbbekre hivatkozásra; mindemellett az olvasó figyelmébe ajánljuk a `ggplot2` [119] és `lattice` [98] csomagokat, melyek hatékony használata bár komolyabb felkészülést igényel, képességeik messze túlszárnyalják az alapvető diagramtípus-megvalósításokat.

5.1. táblázat. Diagramtípusok és R függvények csomagok

Diagramtípus	**függvény**	**csomag**
oszlop, szintvonal, szórás, doboz, hisztogram, mozaik	`barplot, contour, plot, boxplot, hist, mosaicplot`	`graphics`
idősor, hőtérkép	`plot.ts, heatmap`	`stats`
fluktuáció	`fluctile`	`extracat`
párhuzamos koordináták	`parcoord`	`MASS`
szórás-mátrix	`splom`	`lattice`

5.5. Interaktív statisztikai grafika

A számítógéppel megvalósított statisztikai adatvizualizáció lehetőséget teremt arra, hogy egy adatkészlet különböző nézeteivel a felhasználó interakcióba lépjen és ezeknek az interakcióknak *kihatása legyen a többi nézetre*. Az interaktív statisztikai vizualizáció során az eszközök által jellemzően támogatott interakciókat a következőképpen kategorizálhatjuk [110]: *a*) lekérdezések (*queries*); *b*) kiválasztás és csatolt kijelölés (*selection and linked highlighting*); *c*) csatolt elemzések (*linked analyses*) és *d*) helyi interakciók.

A soron következő alfejezetek röviden bemutatják a kategóriákat és a legfontosabb interakciókat. Felhívjuk azonban a figyelmet arra, hogy a különböző eszközök által támogatott interakciókról nem áll módunkban teljes, áttekintő képet adni – az olvasónak javasoljuk a `Mondrian` [109, 110] ingyenes, nyílt forráskódú eszköz megismerését és kipróbálását. Az interaktív statisztikai grafikát ma már több, a vállalati szektornak kínált eszköz is támogatja, az ingyenes és nyílt forráskódúak közül azonban egyértelműen a `Mondrian` a legkiforrottabb. Az `iplots` [115] `R` csomag a `Mondrian`hoz nagyon hasonló képességekkel rendelkezik. Meg kell még említenünk a `GGobi`[24] is, mely a `Mondrian`hoz képest többletfunkciókkal is rendelkezik, kezelése azonban nehézkes és szoftvertechnológiai szempontból is elavultnak tekinthető.

5.5.1. Lekérdezések

A lekérdezések az interaktív diagramon megjelenített elemekkel kapcsolatos statisztikai információk megjelenítését jelentik, jellemzően egér segítségével. A megjeleníthető adatok természetesen az ábra által alkalmazott statisztikai transzformációktól és az aktív kijelölésektől függnek; míg egy szórásdiagramon jellemzően „lekérdezhetjük" pl. egy pont koordinátáit vagy egy kijelölés koordináta-intervallumait, addig egy oszlopdiagramon egy kategória elemszámát és az aktív kijelölésbe tartozó elemek számát. A diagramelem-lekérdezések egy speciális esetét mutatja majd az 5.7. ábra, ahol egy részhalmazra illesztett regressziós egyenes paramétereinek lekérdezése látható.

5.5.2. Helyi interakciók

A diagramokkal önmagukban, a többi diagramra nézve mellékhatásmentesen is interakcióba léphetünk. Egyes operációk – pl. objektumok sorrendjének módosítása, skálamódosítások (ide tartozik a nagyítás is) – általánosan megjelennek; mások ábra-specifikusak. A helyi interakciókkal nem foglalkozunk behatóbban; többségükre tekinthetünk úgy, mint a statikus vizualizáció általában rendelkezésre álló paraméterezési lehetőségeinek „interaktívan" elérhető változataira.

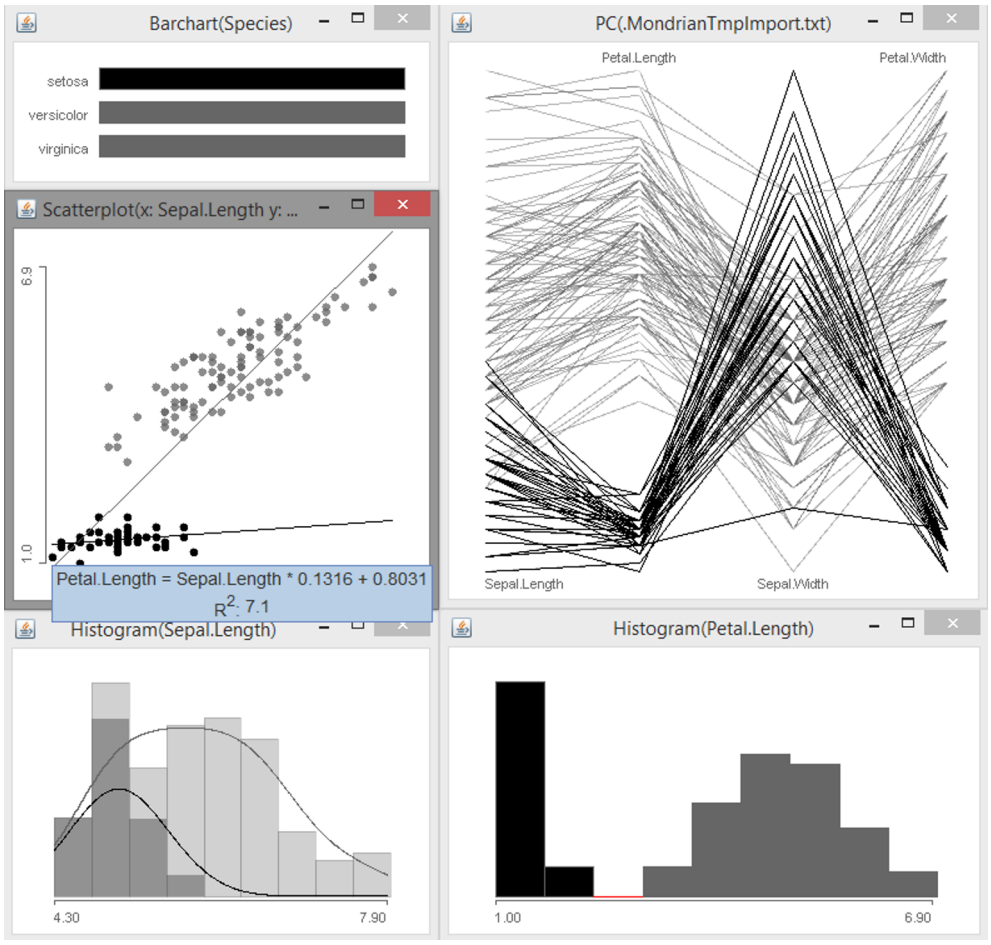
5.5.3. Kiválasztás és csatolt kijelölés

A kiválasztás és csatolt kijelölés az interakció-kategóriák közül a legnagyobb jelentőségű; fogalmazhatunk úgy, hogy ez adja a minőségi különbséget a statikus és az interaktív statisztikai vizualizáció között. Egy adott adatkészleten megjelenített több diagram felfogható úgy, mint egy reláció különböző projekciói (a diagram változóira), majd ezek statisztikai transzformációi. Egy diagramon elemeket (oszlopokat egy oszlopdiagramon, ponthalmazokat egy szórásdiagramon, intervallumot egy dobozdiagramon, ...) jellemzően az egérrel kiválasztva az inverz transzformáció egy „sorhalmazt" határoz meg az eredeti relációban, melynek képe a többi diagramon „kiemelve" vizualizálható.

Erre ad példát az 5.7. ábra a korábban már használt Fisher-féle írisz-adatkészleten, ahol egy, az egérrel „kihúzott" kiválasztó téglalap segítségével a szórásdiagramon választottunk ki pontokat[7]. A kiválasztás motivációja kettős: egyrészt ezek a pontok láthatóan egy elkülönülő klasztert alkotnak a szórásdiagramon, másrészt a kiválasztott csoportra a csészelevél- és sziromhosszúság közötti regressziós kapcsolatot érdemes lenne megvizsgálni. Így ezt a részhalmazt és kapcsolatát a többi megfigyeléssel szeretnénk mélyebben elemezni.

Az adatkészlet egyetlen kategorikus változója, a faj oszlopdiagramján a csatolt kijelölésből rögtön leolvasható, hogy a kiválasztás pontosan az *Iris setosa* faj megfigyelt egyedeit fedi. A párhuzamos koordinátákról leolvasható, hogy ez a faj a sziromhosszúsághoz hasonlóan önmagában a sziromszélességben is szeparálódik a másik kettőtől. Ez a csészelevél

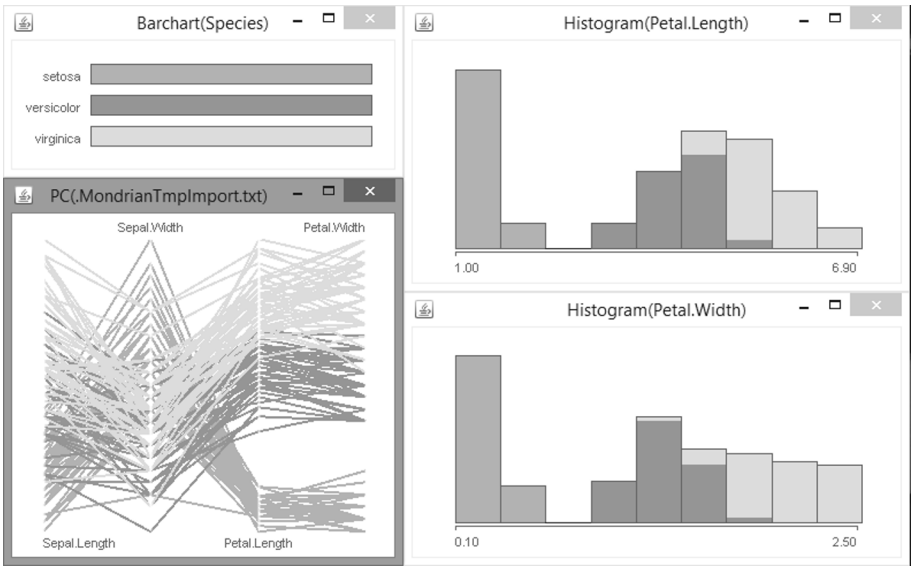
[7] Az eszközökben alapértelmezett esetben a kijelöléseket és a csatolt kiemeléseket karakteres, pl. vörös szín jelöli; a szürke megjelenítés – fekete kiemelés párosra az ábra nyomtatásbaráttá tétele érdekében volt szükség.



5.7. ábra. Kiválasztás és csatolt kijelölés, csatolt elemzés és lekérdezés a `Mondrian`*ban*

jellemzőire nem igaz; felismerhető továbbá, hogy csészelevél-szélesség tekintetében egy megfigyelt egyed a fajon belül kiesőnek (*outlier*) tekinthető. Egyértelmű magas korreláció változópárok között nem olvasható le az adott tengely-permutációnál.

A kiválasztás és csatolt kijelölés egy változatának is tekinthető a kategória-vezérelt színezés, mint interakció (*color brush*). A kategória-vezérelt színezést olyan diagramokról indíthatjuk, melyek kategóriákat jelenítenek meg a vizualizációs primitívként; ilyenek pl. az oszlopdiagram, a mozaik-diagram, de a hisztogram is, ahol az egyes intervallumok oszlopai értelmezhetők kategóriaként. A színezés a kezdeti diagram minden eleméhez – az említett példákon rendre az oszlopokhoz, csempékhez és intervallumokhoz (*bin*-ekhez) – egy színt rendel, melyek segítségével színezi a többi diagram elemeit (pl. egy szórásdiagram pontjait egy-egy színnel, vagy egy oszlopdiagram oszlopait több különböző színű oszlopra bontva, a kategóriák számosságával arányos területtel). Az 5.8. ábra az írisz adatkészlet fajok szerinti színezésére mutat példát.



5.8. ábra. Kategóriánkénti színezés a `Mondrian`*ban*

5.5.4. Csatolt analízis

A kiválasztások hatásának nem feltétlenül kell a (kapcsolt) kiemelésekre korlátozódnia; a kiválasztások változásával reaktívan analízisek, illetve statisztikai modellek felállításai is újrafuthatnak. Ezt az elméleti lehetőséget a jelenlegi eszközök azonban általában nem, vagy csak kevéssé használják ki. A csatolt analízis egyfajta megvalósítása a `Mondrian` szórásdiagramokra illeszthető regressziós egyenes, mely a teljes adatkészlet mellett a mindenkor aktuális kiválasztáson is azonosításra és feltüntetésre kerül. A regressziós modell paraméterei „lekérdezéssel" meg is tekinthetőek (lásd az 5.7. ábra szórásdiagramját).

5.6. Összefoglalás

Az elemzendő adatok „megértésének” és az alapvető összefüggések megsejtésének egyik legfőbb eszköze a statikus és interaktív statisztikai grafika, illetve vizualizáció. A fejezet áttekintést nyújtott ezek legáltalánosabb módszereiről. Nem eshetett szó azonban az adatvizualizáció olyan, jelenleg is aktív kutatás alatt álló területeiről, mint a dinamikus grafika, vagy a szakterületi tudás és statisztikai jellemzők által vezérelt EDA. A további elmélyülést segítendő az érdeklődő olvasó figyelmébe ajánljuk különösen [51] befoglaló könyvét.

6. fejezet

Monte-Carlo-módszerek, Bayesi modellátlagolás, Bayesi predikció

Egy jelenség matematikai modellezése során gyakran felmerülő probléma az alkalmazott modell pontossága (komplexitása) és számítási szempontból való kezelhetősége közti egyensúly megtalálása. Az ideális eset természetesen az, amikor a modell helyesen írja le az ábrázolni kívánt jelenséget, és emellett egzakt módon hajthatóak végre benne a szükséges számítások. Mivel az egzakt módon kezelhető modellek használatát számos szituáció gátolhatja (pl. nem ismert olyan modell, amely megfelelően írja le az adott jelenséget; egy megfelelően pontos modell túlságosan számításigényes lenne, vagy akár nem is rendelkezne zárt megoldással), a rendelkezésre álló számítási kapacitás növekedésével a közelítő módszerek alkalmazása mind nagyobb teret nyert.

Az ilyen, statisztikai alapú, közelítő jellegű módszerek alkalmazásakor tipikusan két fő feladat áll elő: (1) a *modellezés* során meg kell alkotni az adott jelenség leírását; (2) a *következtetés* során a modell alapján kell kiszámítani a tudni kívánt mennyiségeket. Ezek a módszerek az alábbi két fő csoportra oszthatók:

- A *maximum likelihood* módszerek a $P(D|M)$ mennyiséget (azaz az adott D megfigyeléseknek a keresett M modell szerinti feltételes valószínűségét –M likelihoodját) maximalizáló modellt igyekeznek megkeresni, majd a következtetést ez alapján az egyetlen modellpéldány alapján végzik.

- A *bayesi* módszerek során a megfigyelések alapján történő tanulás eredménye egy $\mathbf{P}(M|D)$ a posteriori (utólagos) eloszlás; a következtetés az e fölötti integrál/szumma eredményeként adódik.

Ebben a fejezetben ez utóbbi, bayesi módszerekkel foglalkozunk, pontosabban az ezekhez kapcsolódó *Monte-Carlo-módszerekkel*, amelyek a fent említett integrál/szumma kiszámítására adnak közelítő megoldást. Az alapfeladat tehát a P (X modellek tere feletti eloszlás) felett kiszámítani egy f (X modelleken értelmezett) függvény várható értékét:

$$E_P[f(X)] = \int_\chi f(X)P(X)dX \tag{6.1}$$

$$\approx \frac{1}{m}\sum_{i=1}^{m} f(X_i). \tag{6.2}$$

Bár felmerülhet, hogy egy ilyen integrál értékét valamilyen numerikus módszer alkalmazásával közelítsük, a fejezetben kizárólag a 6.2 alakú, a P eloszlás mintavételezésén alapuló módszerekkel foglalkozunk.

A fejezet további részeinek felépítése a következő. A 6.1 rész a Monte-Carlo-integrálás egyszerűbb alapeseteivel foglalkozik. A 6.2 rész a Markov-láncokról ad áttekintést, amely megalapozza a Metropolis–Hastings-algoritmus bevezetését a 6.3 részben; a 6.4 és 6.5 a fontosabb alesetekkel, illetve a módszer konvergenciájával, konfidenciájával foglalkoznak. A 6.6 rész a Bayes-hálókkal kapcsolatos alkalmazási példákkal foglalkozik.

A fejezet anyagának részletes tárgyalása megtalálható például az [54] vagy a a [93] könyvekben.

6.1. Monte-Carlo-integrálás

A Monte-Carlo-integrálás során tehát az

$$E_P[f(X)] \approx \frac{1}{m}\sum_{i=1}^{m} f(X_i) = \bar{f_m} \tag{6.3}$$

közelítés kiértékelése a cél, ahol $\{X\}_{i=1}^m$ elemei P-ből sorsolt független minták. Ekkor a nagy számok erős törvénye alapján $\bar{f_m}$ majdnem biztosan tart $E_P[f(X)]$-hez, illetve, ha $E_P[f^2(X)]$ véges, akkor a variancia

$$var(\bar{f_m}) = \frac{1}{m}\int_\chi (f(X) - E_P[f(X)])^2 P(X)dX \tag{6.4}$$

$$\approx \frac{1}{m^2}\sum_{i=1}^{m}(f(X_i) - \bar{f_m})^2 = v_m \tag{6.5}$$

alakban közelíthető, ami miatt nagy m-re az $\frac{\bar{f_m} - E_P[f(X)]}{\sqrt{v_m}}$ mennyiség megközelítőleg $\mathcal{N}(0,1)$ eloszlású, ami alapján a konvergenciára vonatkozó teszt, illetve ehhez kapcsolódó konfidencia értékek konstruálhatók.

6.1.1. Közvetlen mintavételezés

A fentiek alapján a legközvetlenebb megoldás a P eloszlás direkt mintavételezése lenne. Ez tipikusan két lépcsőben történhet:

(1) Sorsolunk egy $U \sim \mathcal{U}_{[0,1]}$ uniform eloszlású változó-értéket[1].

(2) Ekkor az $X = P^{-1}(U)$ változó a P eloszlást fogja követni, ahol P^{-1} a P eloszlásfüggvény általánosított inverze[2].

Valós problémák esetén a fenti módszer sajnos szinte sosem alkalmazható, mivel a felmerülő eloszlásfüggvények nem ismertek a kellő mértékben. A következő alfejezetekben olyan módszereket tekintünk át, amelyek a P eloszlás csekélyebb ismerete esetén is alkalmazhatók.

6.1.2. Elutasító mintavételezés

Az elutasító mintavételezés alapötlete a mintavételezendő $p(x)$ sűrűségfüggvény következő lehetséges átírásán alapul:

$$p(x) = \int_0^{p(x)} du, \tag{6.6}$$

ez alapján p az $(X, U) \sim \mathcal{U}\{(x, u) : 0 < u < p(x)\}$ együttes eloszlás X-re vetített marginálisaként jelenik meg. Ez alapján az *elutasító mintavételezés* a következő séma szerint történhet:

(1) Generáljunk le egy (X, U) egyenletes eloszlású változóérték-párt.

(2/a) Ha $U < p(X)$, elfogadjuk X-et, mint p-ből sorsolt értéket.

(2/b) Ha $U \geq p(X)$, eldobjuk X-et.

A megtartott X értékek így valóban a P eloszlást fogják követni.

Tehát az alap algoritmusunk során egy téglalap alakú tartományból sorsolunk pontokat, amelyek közül azokat (azok x-koordinátáját) fogjuk elfogadni, amelyek a szimulálandó sűrűségfüggvény alá esnek. Továbbfejlesztve az algoritmust megtehető, hogy a sorsolási pontokat adó tartomány ne téglalap, hanem egy megfelelő q függvény által meghatározott $\mathcal{L} = \{(y, u) : 0 < u < q(y)\}$ terület legyen.

Ebben az esetben az is megoldható, hogy nem véges maximumú és/vagy tartójú p-ből sorsoljunk; q-nak pedig két triviális feltételt kell teljesítenie: (1) $\forall x \in \text{supp}(p) : q(x) \geq p(x)$ és az egyenletes sorsolásnak lehetségesnek kell lennie $\mathcal{L}$-en.

A q függvény megválasztásánál a következő praktikus szempontokat kell figyelembe venni:

- q-nak a számítási komplexitás szempontjából egyszerűbben kezelhetőnek kell lennie p-nél, hisz a módszer alapmotivációját p bonyolultsága adta.
- Minél távolabb van q p-től, annál több generált mintát kell eldobnunk, tehát p és q közelsége a hatékonyság szempontjából alapvető.

[1] A véletlenszám-generálás algoritmikus részleteinek széles körű irodalma létezik, amit azonban itt nem tárgyalunk, tekintettel, hogy valós problémák esetén maga a közvetlen mintavételezés is inkább csak elméleti lehetőség.

[2] A P függvény általánosított inverzét a $P^{-1}(u) = \inf\{x : P(x) \geq u\}$ kifejezéssel definiáljuk.

6.1.3. Fontossági mintavételezés

Az előző alfejezet alapötletét (hogy egy megfelelő, kezelhető sűrűségfüggvény felhasználásával sorsoljunk tetszőleges eloszlásból) továbbvive juthatunk a *fontossági mintavételezéshez*, felhasználva a várható érték formulájának alábbi felírását:

$$E_P[f(X)] = \int_\chi f(x)\frac{p(x)}{q(x)}q(x)dx \tag{6.7}$$

$$\approx \frac{1}{m}\sum_{i=1}^{m}\frac{p(X_i)}{q(X_i)}f(X_i). \tag{6.8}$$

6.8 ugyanúgy konvergálni fog, mint 6.2, függetlenül q-tól, feltéve, hogy $\mathrm{supp}(q) \supseteq \mathrm{supp}(p)$. Maga a formula már sugallja az alkalmazandó algoritmust: a várható érték számításához elégséges a q sűrűségfüggvényből értékeket sorsolnunk, majd az ezeken számított $f(X_i)$ függvényértékekre a $\frac{p(X_i)}{q(X_i)}$ („fontossági") súlyokat alkalmaznunk.

Bár látható, hogy gyakorlatilag bármilyen q sűrűségfüggvény alkalmas lehet, a becslés

$$E_q\left[f^2(X)\frac{p^2(X)}{q^2(X)}\right] = E_p\left[f^2(X)\frac{p(X)}{q(X)}\right] = \int_\chi f^2(x)\frac{p^2(x)}{q(x)}dx \tag{6.9}$$

varianciája csak akkor lesz véges, ha a fenti integrál véges, vagyis p/q-nak korlátosnak kell lennie.

A fentiek alapján a következő (meglehetősen szigorú) elégségességi feltételek adhatók:

(a) $\forall x \in \chi : \frac{p(x)}{q(x)} < M$ és $\mathrm{var}_p(f) < \infty$; vagy

(b) χ halmaz kompakt, és $\forall x \in \chi : p(x) < P, q(x) > \epsilon$.

A fentiek mellett egy másik lehetőség, hogy a becslésben az m osztót a súlyok

$$\sum_{i=1}^{m} p(x_i)/q(x_i)$$

összegére cseréljük, vagyis:

$$\bar{f_m} = \frac{\sum_{i=1}^{m} f(x_i)\frac{p(x_i)}{q(x_i)}}{\sum_{i=1}^{m}\frac{p(x_i)}{q(x_i)}}. \tag{6.10}$$

Mivel $\sum_{i=1}^{m} p(x_i)/q(x_i) \to 1$, ha $m \to \infty$, a nagy számok erős törvénye szerint a 6.10-beli $\bar{f_m}$ is konvergálni fog $E_p[f(X)]$-hez.

6.2. Markov-láncok

A fejezet eddigi részében olyan eseteket vizsgáltunk, amikor a szimulációhoz szükséges minták közvetlenül generálhatók voltak valamilyen (akár segéd-) eloszlásból. Ennél sajnos

tipikusabb az a helyzet, amikor nem áll rendelkezésre olyan eloszlás, amelyből közvetlenül tudnánk független mintákat generálni, ezért a következő alfejezetekben az előző értéktől függő módon valószínűségi változók sorozatait előállítani képes Markov-láncok alapvető tulajdonságait tekintjük át, illetve, hogy a Metropolis–Hastings-algoritmus segítségével ezek hogyan használhatók fel az ún. Markov-lánc Monte-Carlo-szimulációkban (Markov chain Monte Carlo – MCMC).

Az (X) *Markov-lánc* valószínűségi változók egy olyan $X_1, \dots X_n, \dots$ sorozata, amelyben az X_n jelen állapot a múltbeli állapotoktól csak a megelőző X_{n-1} állapoton keresztül függ, azaz:

$$\forall x_n, x_{n-1}, n : P(X_n = x_n | X_{n-1} = x_{n-1}, \dots X_1 = x_1) = P(X_n = x_n | X_{n-1} = x_{n-1}). \quad (6.11)$$

Ha a fenti P valószínűségek függetlenek n-től, akkor (X)-et *homogénnek nevezzük*; a továbbiakban csak ilyen homogén Markov-láncokkal foglalkozunk. Ebben az esetben a $P(x, A)$ átmenetfüggvény, vagy -kernel a következőképpen definiálható:

- $\forall x \in S : P(x..)$ eloszlás S felett, ahol S az (X) által felvehető állapotok halmaza.
- $\forall A \subset S$-re az $x \mapsto P(x, A)$ függvény kiszámítható.

Véges S esetén a $P_{xy} = P(X_n = y | X_{n-1} = x)$ elemek egy P átmenetmátrixba rendezhetők, amely alapján egyszerűen levezethető, hogy $P^{(m)}(x, y) = P^m(x, y)$, vagyis annak a valószínűsége, hogy az $x \to y$ átmenetet pontosan m lépés alatt tegyük meg, a P mátrix m-edik hatványának megfelelő elemeként számítható. Hasonló megfontolás alapján adódik az ún. *Chapman–Kolmogorov-egyenlet*:

$$P^{(m+n)}(x, y) = \sum_{z \in S} P^{(m)}(x, z) P^{(n)}(z, y). \quad (6.12)$$

A továbbiakban egy futás (állapotátmenet-sorozat) során τ_A-val jelöljük az (X) lánc *megállási idejét*, vagyis azt a legkisebb lépésszámot, amely alatt az A halmaz egy elemére lépett, illetve η_A-val az A-n való áthaladások számát:

$$\tau_A = min\{n \geq 1; X_n \in A\} \quad (6.13)$$

$$\eta_A = \sum_{n=1}^{\infty} I_A(X_n), \quad (6.14)$$

ahol I_A az A halmazba való tartozás indikátorfüggvénye (amely tehát 1, ha $X_n \in A$ és 0 különben).

Visszatérőség. Az $x \in S$ állapot *visszatérő*, ha a láncot onnan indítva, a valószínűség, hogy x-et újra meglátogatjuk 1 ($P(\tau_x < \infty | x) = 1$), illetve *nem visszatérő* (vagy másképpen *tranziens*), ha a vissza nem térés valószínűsége nem nulla ($P(\tau_x < \infty | x) < 1$).

Ha egy visszatérő x állapot esetén, a visszatérési idő várható értéke véges (azaz $\sum_{n=1}^{\infty} nP(\tau_x = n | x) < \infty$), az x állapotot *pozitív visszatérőnek*, ellenkező esetben *null visszatérőnek* nevezzük.

Irreducibilitás. Az (X) lánc egy fontos tulajdonsága, hogy az egyes állapotok egymásból elérhetők-e. Ha $\forall x, y \in S : P(\tau_y < \infty | x) > 0$, vagyis bármely két állapot között létezik véges hosszú, nem nulla valószínűségű átmenet, a láncot *irreducibilisnek* nevezzük.

Megmutatható, hogy irreducibilis láncok esetén a visszatérőség és a pozitív visszatérőség tulajdonságok az összes $x \in S$ állapotra együtt lesznek igazak vagy hamisak.

Stacionárius eloszlás. Az S állapottér feletti π eloszlást *stacionárius* eloszlásnak nevezzük, ha $\pi = \pi P$, vagyis, ha a lánc egy adott pillanatban az adott π eloszlással tartózkodik lehetséges állapotaiban, akkor a láncnak ez a marginális eloszlása az (összes) következő állapotban is ugyanaz lesz.

Megmutatható, hogy egy irreducibilis Markov-lánc akkor és csak akkor pozitív visszatérő, hogyha létezik π stacionárius eloszlása, és az:

$$\pi(y) = \lim_{n \to \infty} \frac{\sum_{k=1}^{n} P^k(x, y)}{n}. \tag{6.15}$$

Határérték-tételek. Egy $x \in S$ állapot d_x *periódusa* a $\{n \geq 1 : P^n(x, x) > 0\}$ halmaz elemeinek legnagyobb közös osztója. Ha a lánc irreducibilis, minden állapotának periódusa azonos, ha $d = 1$ *aperiodikus* állapotról (láncról) beszélünk, ellenkező esetben *periodikusról*. Egy pozitív visszatérő, aperiodikus állapotot (láncot) *ergodikusnak* nevezünk.

Ha a lánc aperiodikus, $\forall x, y \in S : \lim_{n \to \infty} P^n(x, y) = \pi(y)$, illetve, ha a lánc irreducibilis és ergodikus, akkor $\forall x \in S : \lim_{n \to \infty} ||P^n(x, .) - \pi(.)|| = 0$, ahol $||\xi_1 - \xi_2|| = \sup_{A \subset S} |\xi_1(A) - \xi_2(A)|$.

A nagy számok törvényének Markov-láncokra alkalmazható analógja az *ergodikus tétel*, amely szerint, ha a lánc ergodikus, akkor tetszőleges f függvényre

$$\hat{f}_n = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \to E_\pi[f(X)] \tag{6.16}$$

majdnem biztosan, ha $n \to \infty$.

Egy lánc *geometrikusan ergodikus*, ha létezik $0 \leq \lambda < 1$ és M integrálható függvény, hogy

$$\forall x \in S || P^n(x, .) - \pi(.)|| \leq M(x) \lambda^n; \tag{6.17}$$

ahol a legkisebb, a feltételt kielégítő λ-t a konvergencia *sebességének* hívjuk. Ha $M(.)$ értéke nem függ x-től, az ergodicitás *uniform*.

A Markov-láncokra megfogalmazható központi határeloszlás-tétel a következő. Ha az (X) lánc uniform (geometrikusan) ergodikus, és $E_\pi[f^2(X)] < \infty$ ($\exists \epsilon > 0 : E_\pi[f^{(2+\epsilon)}(X)] < \infty$), akkor a

$$\overline{f} = E_\pi[f(X)] \tag{6.18}$$

$$\sigma^2 = var_\pi[f(X)] \tag{6.19}$$

$$\tau^2 = \sigma^2 + 2 \sum_{i=1}^{\infty} E_\pi[(f(X_0) - \overline{f})((f(X_k) - \overline{f})] \tag{6.20}$$

jelölések mellett, ha τ^2 létezik és véges, akkor

$$\sqrt{n}\frac{\hat{f}_n - \overline{f}}{\tau} \to \mathcal{N}(0,1) \tag{6.21}$$

eloszlásban, ha $n \to \infty$.

Reverzibilitás. Egy lánc *reverzibilis*, ha igaz rá, hogy

$$\forall x, y \in S : \pi(x)P(x,y) = \pi(y)P(y,x), \tag{6.22}$$

vagyis a lánc stacionárius állapotában annak a valószínűsége, hogy egy adott átmenet során x-ből lépünk y-ba, ugyanannyi, mint annak, hogy y-ból x-be. A reverzibilitás hasznos tulajdonság, mivel, ha egy irreducibilis lánchoz létezik egy a fenti egyenletet kielégítő π eloszlás, akkor a lánc pozitív visszatérő lesz π stacionárius eloszlással; vagyis egy Markov-lánc konstruálása egy π kívánt stacionárius eloszlással a megfelelő $P(.,.)$ megkeresésére redukálódik, ami pedig (mint azt 6.3 alfejezetben látni fogjuk) mindig lehetséges.

6.3. Metropolis–Hastings-algoritmus

A *Markov-lánc Monte-Carlo (MCMC)* szimulációk alapötlete a p eloszlás alapján történő számításokra a következő:

(1) Hozzunk létre egy ergodikus Markov-láncot (X_n), amelynek a stacionárius eloszlása p.

(2) Közelítsük az $E_P[f(X)]$ mennyiséget az (X_n) lánc által érintett állapotok alapján:

$$E_P[f(X)] = \int_{\mathcal{X}} f(x)p(x)dx \approx \frac{1}{m}\sum_{i=1}^{m} f(X_i) = F_m. \tag{6.23}$$

A lánc ergodicitásából következik, hogy (elméletben) annak X_0 kiindulási állapota bármi lehet. Bár első pillantásra a korábban ismertetett eljárások (v.ö. fontossági mintavételezés, elutasító mintavételezés) direktebb, és így kecsegtetőbb megoldást kínálnak, valójában az, hogy az MCMC eljárások során elégséges a korábbi állapottól függő módon előállítani a felhasznált mintákat, jelentős könnyítést jelent, aminek segítségével az MCMC módszerek az előzőeknél hatékonyabban képesek megbirkózni az olyan praktikus estekben jelentkező problémákkal, mint például amit a mintavételezett tér nagy dimenziószáma jelent.

Fontos tehát kiemelni, hogy a 6.23 képlet alkalmazásakor az összes érintett minta felhasználható; nem szükséges „független” mintákat előállítani, például oly módon, hogy N darab MCMC futásnak csak az utolsó állapotait használjuk fel.

Ebben az alfejezetben tehát az MCMC módszerek legáltalánosabb megvalósítását a *Metropolis–Hastings-algoritmust* tekintjük át, majd a következőkben sorra vesszük a módszer különféle specializált aleseteit.

6.3.1. Definíció. *Metropolis–Hastings-algoritmus.* Adott a π cél-sűrűségfüggvény, valamint konstruálnunk kell egy $q(y|x)$ feltételes sűrűségfüggvényt, amelyre a következő (esetleg informális) feltételeknek kell teljesülniük:

(1) $q(.|x)$-nek könnyen szimulálhatónak kell lennie.

(2/a) Vagy explicite, egy multiplikatív (x-től független) konstans erejéig ismertnek kell lennie,

(2/b) vagy szimmetrikusnak kell lennie ($\forall x, y : q(y|x) = q(x|y)$).

(3) A $\frac{\pi(x)}{q(y|x)}$ mennyiségnek egy multiplikatív konstans erejéig ismertnek kell lennie.

Ekkor a *Metropolis–Hastings-algoritmus* a következő lépésekkel halad (feltéve, hogy a lépés elején az X_t állapot adott):

(1) Generáljuk le az $Y_t \sim q(y|x_t)$ *javasolt* mintát.

(2) Számítsuk ki a $\rho(X_t, Y_t)$ *elfogadási* valószínűséget, mint: $\rho(x, y) = \min\left\{\frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}, 1\right\}$.

(3) Legyen $X_{t+1} = Y_t$ $\rho(X_t, Y_t)$ valószínűséggel, illetve $X_{t+1} = X_t$ máskülönben.

Bár a fenti algoritmus meglehetősen kevés megkötést tesz q-val kapcsolatban, a helyes működéshez fontos, hogy a π sűrűségfüggvény tartóját teljesen lefedje, azaz, hogy (1) ne legyen olyan $A \subset \mathrm{supp}(\pi)$, ahol $\int_A \pi(x)dx > 0$, de $\forall x \in \mathcal{X} : \int_A q(y|x)dy$, illetve, hogy viszont nem egybefüggő tartó esetén bármely részből elérhető a többi.

A fenti feltétel teljesülése mellett már belátható, hogy a vizsgált láncra teljesül az ún. *részletes egyensúly* (detailed balance) feltétel, ami elégséges (de nem szükséges) feltétele a konvergenciának, illetve annak, hogy a stacionárius eloszlás π legyen.

A leírt algoritmus szerinti átmenet-kernel az alábbi lesz:

$$K(x, y) = \rho(x, y)q(y|x) + (1 - r(x))\delta_x(y), \tag{6.24}$$

ahol $r(x) = \int \rho(x, y)q(y|x)dy$, δ_x pedig az x-központú Dirac-delta függvény. Ezek alapján már belátható, hogy

$$\rho(x, y)q(y|x)\pi(x) = \rho(y, x)q(x|y)\pi(y), \tag{6.25}$$
$$(1 - r(x))\delta_x(y)\pi(x) = (1 - r(y))\delta_y(x)\pi(y), \tag{6.26}$$

ami már biztosítja a részletes egyensúly tulajdonságot.

Azt, hogy a Metropolis–Hasting-algoritmus által generált Markov-lánc valóban a stacionárius eloszlásához tart, belátható a következők alapján:

(1) Az *elutasítás* események (amikor tehát $X_{t+1} = X_t$) valószínűsége nem nulla, emiatt a lánc *aperiodikus*.

(2) q megfelelő megválasztása ($\forall x, y \in \mathrm{supp}(\pi) : q(y|x) > 0$) biztosítja az *irreducibilitást*.

Ezekből pedig az ergodicitásra vonatkozó tétel alapján levezethető a konvergencia ténye.

Triviális alesetek. A Metropolis–Hastings-algoritmus két legegyszerűbb alesete a (1) *véletlen bolyongás Metropolis–Hastings-algoritmus*, amikor $q(y|x)$ javaslati eloszlás x-nek csak valamilyen szűkebb környezetében nem nulla, illetve q értéke csak a két állapot valamilyen távolságától függ: $q(y|x) = q(|y - x|)$; valamint a (2) *független mintavételező Metropolis–Hastings-algoritmus*, amikor q csak a javasolt állapottól függ, a kiindulásitól nem: $q(y|x) = q(y)$.

6.4. A Metropolis–Hastings-algoritmus alesetei

Az előző alfejezetben bemutattuk a Metropolis–Hastings-algoritmus alapvető felépítését és működését; ebben pedig sorra veszünk néhányat a legjelentősebb alesetei közül.

6.4.1. Gibbs-mintavételezés

Bár a Gibbs-mintavételezés koncepciója történetileg megelőzte a Metropolis–Hastings-algoritmust, valamint a kifejlesztéséhez vezető motiváció is más területről (a képfeldolgozásról) ered, mégis tekinthető ez utóbbi egy speciális alesetének.

Gibbs-mintavételezés során az alkalmazott modell a következő:

(1) A θ paraméterezés feletti $\pi(\theta)$ eloszlást kívánjuk mintavételezni, amelyről feltételezhető, hogy önmagában nem, vagy csak nehezen kezelhető.

(2) θ maga is egy különálló mennyiségekből képzett vektor: $\theta = (\theta_1, \dots \theta_n)$.

(3) Ismert és könnyen kezelhető az összes $\pi_i(\theta_i) = \pi(\theta_i|\theta_1, \dots \theta_{i-1}, \theta_{i+1}, \dots \theta n)$ feltételes eloszlás.

Ennek megfelelően a minták generálása a következőképpen halad:

(1) Inicializáljuk a láncot valamilyen paraméterérték-halmazzal: $\theta^{(0)} = (\theta_1^{(0)}, \dots \theta_n^{(0)})$.

(2) Az $i + 1$-edik mintát az i-edik alapján az alábbiak szerint állítjuk elő:

$$\theta_1^{(i+1)} \sim \pi(\theta_1|\theta_2^{(i)}, \theta_n^{(i)}) \tag{6.27}$$

$$\vdots$$

$$\theta_k^{(i+1)} \sim \pi(\theta_k|\theta_1^{(i+1)}, \dots \theta_{k-1}^{(i+1)}, \theta_{k+1}^{(i)}, \dots \theta_n^{(i)}) \tag{6.28}$$

$$\vdots$$

$$\theta_n^{(i+1)} \sim \pi(\theta_n|\theta_1^{(i+1)}, \dots \theta_{n-1}^{(i+1)}), \tag{6.29}$$

vagyis egymás után végiglépkedünk az egyes paramétereken, és a soron következőnek a többi aktuális értéke alapján sorsolunk új aktuális értéket.

(3) Ha a $\theta^{(i)}$ elemek elérték a konvergenciát, akkor valóban a π eloszlást követik; vagyis egy kezdeti burn-in szakasz után a kapott minták (mint azt a Markov-láncoknál láttuk) felhasználhatók.

A fenti séma alapján könnyen belátható, hogy a Gibbs-mintavételező egy homogén Markov-láncot alkot (amely ugyan nem reverzibilis[3]), és viszonylag egyszerűen levezethető az is, hogy a stacionárius eloszlása valóban π lesz.

Bejárási stratégiák. A fenti sémában θ elemein minden lépésben egy adott sorrendben $(1, \dots n)$ mentünk végig, valójában bármely olyan lépéssorozat megfelelő, amely biztosítja, hogy egy végtelen hosszú futás során minden komponenst végtelen gyakran ($E[\eta_{\theta_i}] = \infty$) felkeresünk.

Ennek megfelelően szintén konvergens Gibbs-mintavételezőt kapunk, ha minden lépésben $1, \dots n$-ből egyetlen indexet sorsolunk (úgy, hogy minden index pozitív valószínűséggel kerül kisorsolásra), és csak azt az egy komponenst mintavételezzük újra. Szintén egy lehetséges séma, ha a komponensek bejárási sorrendjét minden lépésben újrasorsoljuk.

A Gibbs-mintavételező reverzibilissé is tehető a korábban már említett módon, vagyis, ha minden lépésben $2n$ komponens-módosítást végzünk, az $1, \dots n, n, \dots 1$ sorrendben.

6.4.2. Többláncos MCMC

Az MCMC szimulációk egy központi kérdése a konvergencia, azaz, hogy valóban sikerül-e a π céleloszlást mintavételeznünk. Gyakori probléma a teljes állapottér bejárásának kérdése, azaz, hogy a javasolt lépésekkel eljutunk-e „mindenhova". Mivel a lépések elfogadása a π által az adott állapotokban felvett értéktől függ, annak jellege is befolyásolhatja a konvergenciát: egy olyan π esetén, amely több lokális maximumból és a köztük lévő nullközeli területekből áll, az algoritmus könnyen „beragadhat".

Egy lehetséges megoldás erre a problémára a *szimulált lehűtésnél* is alkalmazott technika bevetése: az eredeti $\pi(\theta)$ eloszlást egy T hőmérséklet-paraméter értéke szerint a $q \propto \pi^{\frac{1}{T}}(\theta)$ értékekre cseréljük. $T > 1$ értékekre az alkalmazott transzformáció az eredeti π eloszlás „simított" változatát állítja elő, amely így könnyebben bejárhatóvá válik. Ez az ötlet egyetlen lánc használatakor is alkalmazható, hiszen a valódi (π) és az alkalmazott (q) eloszlás aránya ismert, így a minták súlyozott figyelembe vételén alapuló eljárás alkalmazható.

Egy másik lehetőség több MCMC lánc párhuzamos futtatása, különböző, $T_i = 1+\lambda(i-1)$ paraméterekkel. Ekkor mindig az $i = 1$ sorszámú (tehát „fűtetlen") láncot mintavételezzük, viszont bizonyos lépésközönként két lánc θ állapotát felcseréljük a következők szerint: (1) javaslatot teszünk a két felcserélendő lánc sorszámára (ez történhet egy a lehetséges (i, k) párok feletti egyenletes eloszlás szerint, de tipikus az is, hogy kikötjük, hogy $k = i+1$ – ez a csere elfogadásának valószínűségét növeli); (2) a csere elfogadásának valószínűsége:

$$\alpha_{ik}(\theta_i^{(j)}, \theta_k^{(j)}) = \frac{\pi_i(\theta_k^{(j)})\pi_k(\theta_k^{(j)})}{\pi_i(\theta_i^{(j)})\pi_k(\theta_k^{(j)})}. \tag{6.30}$$

[3]A Gibbs-mintavételező könnyen reverzibilissé tehető, ha egy lépés nem n, hanem $2n$ allépésből áll, amelyek során az $1, \dots n, n, \dots 1$ sorrendben megyünk végig a paramétereken és adunk nekik új értéket a π_i eloszlásoknak megfelelően.

Az eljárás alapötlete jól látható: a magas hőmérsékletű láncok könnyebben mozognak, így jobban bejárják a teljes teret; az állapotcseréken keresztül a „fűtetlen" lánc is eljuthat ezekbe az állapotokba; viszont mivel az az eredeti π céleloszlást használja (és maga a csere is e szerint kerülhet csak elfogadásra), a nyert minták is valóban π-t követik.

Az eljárás hátránya is nyilvánvaló: m párhuzamos lánc használatával a számítási igény is az m-szeresére nő.

6.4.3. Reversible jump MCMC

Az eddig vizsgált módszerek egy közös (implicit) alapfeltevése volt, hogy a vizsgálni kívánt eloszlás (a Θ modellparaméterek tere) struktúráját jól ismerjük: ennek a feltevésnek a kihasználásával voltunk képesek végrehajtani akár mintavételezést, akár az ehhez alkalmazott segédfüggvények megkonstruálását. A *reversible jump MCMC* módszerek az olyan eseteket próbálják kezelni, amelyekben ez a feltevés nem igaz, vagyis a vizsgált modelltér nem ismert eléggé ahhoz, hogy a példányait leíró paraméterezések dimenziószámát meg tudjuk adni. Ilyen szituáció alakulhat ki tipikusan modellkonstrukciós vagy -kiválasztási feladatoknál, amikor a rendelkezésre álló tudásunk nem elégséges ahhoz, hogy egyetlen modellosztály alkalmazása mellett köteleződjünk el.

Az ilyen esetekben a teljes modelltér több, különböző dimenziójú altérre bomlik, amelyek között az átlépések nem kezelhetők az eddig megismert módszerek alkalmazásával. A fő probléma tehát a jelen helyzetben a korábbiakhoz képest a különböző dimenziószámú modellpéldányok közötti átlépések kezelése. A Green (1995) által javasolt megoldás a problémára egy segédváltozó bevezetésén alapul, amely segítségével a különböző dimenziójú alterek között egy bijekciót hozunk létre.

Tegyük fel, hogy a Θ_{k_1} és a Θ_{k_2} alterek között akarunk mozogni, ahol $\dim(\Theta_{k_1}) > \dim(\Theta_{k_2})$. Ekkor ha rendelkezésre áll egy T determinisztikus transzformáció, amely Θ_{k_1}-ről képez Θ_{k_2}-re, akkor az ellenkező irányú lépéskor a $\{\theta_{k_1} : T(\theta_{k_1}) = \theta_{k_2}\}$ halmaz elemeiből kell majd választanunk.

A gyakorlatban ilyenkor alkalmazott lépés az, hogy θ_{k_1}-et és θ_{k_2}-t is kiegészítjük egy $u_{k_1} \sim g_1(u_{k_1})$, illetve $u_{k_2} \sim g_2(u_{k_2})$ értékkel, úgy, hogy a $(\theta_{k_2}, u_{k_2}) = T(\theta_{k_1}, u_{k_1})$ transzformáció bijekció legyen. Ebben az esetben az $\mathcal{M}_{k_1} \to \mathcal{M}_{k_2}$ lépés elfogadási valószínűsége

$$min\left(\frac{\pi(k_1, \theta_{k_1})}{\pi(k_2, \theta_{k_2})} \frac{\pi_{21} g_2(u_2)}{\pi_{12} g_1(u_1)} \left|\frac{\partial T(\theta_{k_1}, u_1)}{\partial(\theta_{k_1}, u_1)}\right|, 1\right) \tag{6.31}$$

lesz.

6.5. Konvergencia

Az eddig bemutatott módszerek alapfeladata tehát egy π céleloszlás mintavételezése. Az elméleti eredmények ugyan garantálják is, hogy a π-hez való konvergencia megtörténik, ha az n lépésszám tart a végtelenhez, a valós alkalmazásokban praktikusabb eredményekre van szükségünk.

Ebben az alfejezetben két kérdést vizsgálunk meg vázlatosan: (1) hogyan állapítható meg, hogy a futtatott szimulációk megfelelő eredményeket szolgáltatnak-e (azaz praktikusan, hogy a konvergencia megtörtént-e), illetve (2) milyen lehetőségek vannak a konvergáló láncokból származó minták hatékony felhasználására.

6.5.1. Konvergencia tényének vizsgálata

Általános esetben a fejezetben vizsgált módszerek akkor szolgáltatnak a π céleloszlásból származó mintákat, ha a lánc által megtett lépések száma tart végtelenhez. Praktikus helyzetekben ez nyilván nem egy megvalósítható lehetőség, a konvergencia-diagnosztika feladata tehát annak megállapítása, hogy hány kezdeti lépés (az ún. *burn-in* szakasz) után közelíti a lánc állapotainak eloszlása π-t megfelelően. A következőkben az erre szolgáló statisztikai/empirikus módszereket tekintjük át, tehát azokat, amelyek a lánc által szolgáltatott kimenet vizsgálatával adnak becslést a konvergencia elégséges mivoltára.

Informális módszerek. A konvergencia ellenőrzésének legegyszerűbb módja egy megfelelő, a modelltér állapotai felett értelmezett $f : \theta \mapsto \mathbb{R}$ függvény diagramjának vizsgálatával történhet. Ennek lépései a következők.

Futtassunk n darab egymástól független láncot, majd a vizsgálni kívánt b burn-in lépés után ábrázoljuk az $f(\theta_i^{(b+)})$ értékeket. Ha az egyes futásokhoz tartozó diagramok nem különböztethetők meg egymástól, elfogadhatjuk a konvergencia tényét.

Hasonló módszer alkalmazható egyetlen lánc esetén is: ha a burn-in utáni különböző diagramszakaszok megkülönböztethetetlenek, a lánc konvergensnek tekinthető.

Geweke-mutató. A Geweke által javasolt módszer egy láncon belül vizsgálja az $f(\theta)$ függvény értékeire különböző futási szakaszokon számított ergodikus átlagokat. A módszer szerint: vegyük a b hosszú burn-in szakasz utáni n lépés hosszú szakasz n_a és n_b részeit, ahol $n_a + n_b < n$. Legyen tehát:

$$\overline{\psi}_b = \frac{1}{n_b} \sum_{i=b+1}^{b+n_b} f(\theta^{(i)}) \qquad \overline{\psi}_a = \frac{1}{n_a} \sum_{i=b+n-n_a+1}^{b+n} f(\theta^{(i)}); \tag{6.32}$$

azaz az f függvénynek a vizsgált szakasz elején és végén vett ergodikus átlaga. Ekkor n_a/n és n_b/n fix értéke mellett, ha $n \to \infty$, akkor

$$z_G = \frac{\overline{\psi}_a - \overline{\psi}_b}{\sqrt{\widehat{\mathrm{var}}(\psi_a) + \widehat{\mathrm{var}}(\psi_b)}} \to \mathcal{N}(0,1) \tag{6.33}$$

eloszlásban. Ezek alapján a z_G-re számított nagy értékek kizárják a konvergenciát (bár kis z_G értékből még nem következik a konvergencia ténye). A vizsgált részszakaszok hossza tipikusan: $n_b = 0{,}1n$, illetve $n_a = 0,5n$.

Gelman–Rubin-mutató. Ez az eljárás független futtatások alapján vizsgálja a konvergenciát, az alábbi két mennyiség alapján:

$$B = \frac{n}{m-1}\sum_{i=1}^{m}(\overline{\psi}_i - \overline{\psi})^2 \qquad W = \frac{1}{m(n-1)}\sum_{i=1}^{m}\sum_{j=1}^{n}(\psi_i^{(j)} - \overline{\psi}_i)^2, \tag{6.34}$$

ahol $\psi_i^{(j)} = f(\theta_i^{(j)})$ az i-edik lánc (j)-ben felvett értékére számított érték, $\overline{\psi}_i$ ezen értékek átlaga az i-edik láncon belül, $\overline{\psi}$ az előbbi átlagok átlaga (a láncok felett), m a láncok száma, n pedig a vizsgált szakaszok hossza; ezek alapján B ψ varianciája a láncok között, W pedig a láncokon belül.

Ekkor belátható, hogy az

$$\widehat{R} = \sqrt{\frac{(1-\frac{1}{n})W + \frac{1}{n}B}{W}} \tag{6.35}$$

mennyiségre, ha $n \to \infty$, akkor $\widehat{R} \to 1$, azaz a konvergencia ténye $\widehat{R}$ 1-hez való közelsége alapján becsülhető.

6.5.2. Mintavételezés hatékonysága

A 6.5 alfejezetben vizsgált második kérdés tehát, hogy a szimulációk által szolgáltatott, a későbbi számításokra felhasználandó mintát hogyan lehet/érdemes előállítani. Tipikusan a következő lehetőségek léteznek:

(1) Ha feltételezzük, hogy a lánc b burn-in lépés megtétele után konvergál, vagyis ennyi lépés kell, hogy a lehetséges állapotok eloszlása valóban π legyen, akkor n darab független minta előállítása $b \times n$ lépésben történhet: indítunk n darab egymástól független láncot, és mindegyiknek az utolsó állapota kerül be a mintába.

Nyilván számítási igény szempontjából ez egy nagyon pazarló módszer, tehát más lehetőségeket is meg kell vizsgálni.

(2) A másik véglet, ha az ergodikus átlagokra vonatkozó tétel alapján egyetlen láncot használunk csak, és ebből mintavételezünk n darab mintát a b hosszú burn-in után; így összesen $b + n$ lépést kell végeznünk.

Ennek a módszernek az alkalmazhatósága viszont a láncon belüli autokorrelációtól függ: ha az túl nagy, a minták nem lesznek eléggé függetlenek egymástól, ami viszont n növelését fogja szükségessé tenni.

(3) Az előző pont problémáját kiküszöbölendő, megtehetjük, hogy a burn-in után csak minden k-adik mintát használjuk fel (amelyek között így az autokorreláció jelentősen csökkenthető), azaz összesen $b + k \times n$ lépést teszünk; ami az (1) lehetőséghez képest még mindig jelentős megtakarítást jelent, ha $k \ll b$.

Valójában megmutatható, hogy az így kapott becslés rosszabb, mintha mind a $k \times n$ mintát, amelyet a burn-in után érintünk, figyelembe vennénk; ennek a módszernek tehát csak akkor van valós haszna, ha (pl. memóriakorlátok miatt) a felhasználható minták száma limitált.

(4) Végül egy kompromisszumos megoldás lehet, ha kis számú (l) független láncot indítunk, amelyekből azok konvergenciája után aztán egyenként n/l mintát veszünk; így összesen $l \times b + n$ lépést kell tennünk.

6.6. Alkalmazás Bayes-hálókban

A Bayes-hálók a valószínűségi alapú, bizonytalan tudás ábrázolásának egyik legfontosabb eszközei; a hozzájuk kapcsolódó bayesi következtetési esetek pedig – a paramétertér komplexitása miatt – gyakran igényelik valamilyen Monte-Carlo-módszer alkalmazását. Ebben az alfejezetben az MCMC módszerek Bayes-hálókhoz kacsolódó alkalmazási eseteit tekintjük át.

Bayes-hálók. Egy Bayes-háló egy irányított körmentes gráf (Directed Acyclic Graph – DAG), amelyben az egyes csomópontok egy-egy valószínűségi változót reprezentálnak, a felettük definiált együttes valószínűségi eloszlás pedig a következő egyenlet szerint adódik:

$$P(x_1, \dots x_n) = \prod_{i=1}^{n} P(x_i | \mathrm{Pa}(x_i)), \tag{6.36}$$

vagyis az együttes eloszlás az egyes változóknak a szüleik szerinti feltételes eloszlásainak szorzataként adódik.

Prediktív következtetés. A prediktív következtetés feladata, hogy egy konkrét Bayes-háló (struktúra, valószínűségi paraméterezés páros) esetén $P(\mathbf{T} = \mathbf{t} | \mathbf{E} = \mathbf{e})$ jellegű valószínűségeket meghatározzon ($\mathbf{T}$ az ún. célváltozók, $\mathbf{E}$ pedig az ismert értékű *evidenciák* halmaza). Ugyan a feladat bizonyos speciális eseteire léteznek egzakt algoritmusok, általános esetben valamilyen Monte-Carlo-eljárás bevetése szükséges. A lehetőségek itt a következők:

(1) Ha a hálóban egyetlen evidencia-változó sincs (vagyis egyik változó értéke sincs rögzítve), egy teljes változókonfiguráció legenerálása könnyen megoldható: a topologikus sorrendben végiglépkedünk a változókon, és minden lépésben behelyettesítjük az adott változót a szülei (amelyek ekkor már rendelkeznek hozzárendelt értékkel) által meghatározott feltételes eloszlásból sorsolva.

Az erre építő elutasító mintavételezés ennek megfelelően a következő: (1) az „üres" hálóból generálunk n darab mintát; (2) ezek közül elvetjük azokat, amelyek inkompatibilisek az evidenciákkal; (3) a $P(\mathbf{t}|\mathbf{e})$ valószínűség a megtartott mintákon a célváltozó-konfigurációval kompatibilis minták részarányaként adódik.

(2) Az előző módszer továbbfejlesztése, ha *elutasító mintavételezés* helyett *fontossági súlyozást* alkalmazunk úgy, ha csak olyan alapminta-elemeket generálunk, amelyek az evidenciák megadott értékeivel kompatibilisek, és ezeket a mintákat az evidencia-változók feltételes valószínűségeivel súlyozzuk. Így minden legenerált mintát felhasználhatunk, a $P(\mathbf{t}|\mathbf{e})$ valószínűség pedig nem egy egyszerű számarány, hanem a minták súlyai összegének arányaként adódik majd.

(3) A fenti minta generálására a *Gibbs-mintavételezés* is alkalmazható a következők szerint: (1) a lánc kiinduló állapota a háló változóinak egy az evidenciákkal kompatibilis teljes behelyettesítése; (2) egy Gibbs-lépés során a nem-evidencia változókon lépkedünk végig, és azokat mindig a többi változó aktuális értéke alapján adódó feltételes valószínűség alapján sorsoljuk újra.

Ez a módszer kiküszöböli a *fontossági súlyozás* azon hibáját, hogy az hajlamos nagyon kis valószínűségű (súlyú) minták generálására, amelyek a valószínűség meghatározásában nagy számuk ellenére is kis jelentőséggel bírnak.

Parametrikus következtetés. Parametrikus következtetés esetén a adott megfigyelési adathalmaz mellett keressük bizonyos modellparaméterek vagy -jegyek valószínűségét vagy a posteriori eloszlását. Az itt alkalmazott eljárások tipikusan a lehetséges modellhalmazok feletti Metropolis–Hastings-algoritmust használnak.

Mivel az ilyen modellhalmazok számossága szinte minden praktikus alkalmazási esetre kezelhetetlenül nagy, itt nem cél a modellpéldányok feletti eloszlásban való konvergencia; ehelyett szinte mindig valamilyen, a modell alapján számítható strukturális jegy[4] értékének eloszlását becsüljük a $P(\text{jegy}|\text{modell})$ feltételes valószínűségek alapján.

A két leggyakrabban alkalmazott összeállítás a következő:

Strukturális MCMC. A mintavételezendő eloszlás ebben az esetben a $\pi = P(s|D_n)$, az s struktúrák D_n megfigyelési adatok szerinti feltételes eloszlása, a Q javaslati eloszlás pedig az adott struktúrából egy „lépéssel" elérhető struktúrák feletti egyenletes eloszlás; ahol a lehetséges lépések a következők: (1) új él hozzáadása a struktúrához, (2) meglévő él törlése a struktúrából, és (3) meglévő él megfordítása.

Sorrendi MCMC. Az előzőnél gyorsabb konvergenciával kecsegtet a struktúrák tere helyett a topologikus sorrendek felett futó MCMC eljárás. Ebben az esetben Q a javaslati eloszlás a következő elemi lépésekből adódik: (1) két változó pozíciójának felcserélése a sorrendbe, (2) „vágás": az aktuális sorrendet két diszjunkt szakaszra bontjuk, és a két szakaszt megcseréljük.

[4]Ilyen lehet például az adott célváltozót a háló többi részétől valószínűségi értelemben feltételesen elválasztó ún. *Markov-takaró*.

Mint említettük, a sorrendi-MCMC eljárás előnye a gyorsabb konvergencia (alacsonyabb lépésszám-igény), hátrányai viszont, hogy egy-egy lépés megtételéhez jóval nagyobb számítási igény társul, illetve a mintavételezett jegyek feltételes valószínűségének számítása is jóval bonyolultabb: míg ez a strukturális MCMC esetén egy indikátorfüggvény számításával megoldható, sorrendi MCMC esetén az itt jelentkező számításigény tetemes lehet.

7. fejezet

Bootstrap-módszerek

A statisztikai módszerek által végrehajtott feladatok a legáltalánosabban a következőképpen írhatók le:

- Egy adott (tipikusan ismeretlen) eloszlás szerinti objektumoknak keressük valamilyen tulajdonságát.
- Mivel maga a P eloszlás nem ismert, az $\mathbf{x} = \{x_1, \dots x_n\} \sim P$ minta alapján kell megbecsülnünk a keresett mennyiséget.
- Az adott becslés jóságát is meg kell becsülnünk, például a varianciájára való becslés és/vagy a konfidenciájára vonatkozó kijelentések adásával.

A leggyakoribb példa, a *várható érték* számítása esetén mind magára a keresett értékre, mind annak a *varianciával* becsült *standard hibájára* ugyan egyszerű és jól kezelhető képletek állnak rendelkezésre:

$$E_P(x) \approx \overline{x} = \frac{1}{n} \sum_{i_1}^{n} x_i, \quad \mathrm{se}(\overline{x}) \approx \sqrt{\frac{\sum_{i=1}^{n} \frac{(x_i - \overline{x})^2}{n-1}}{n}}; \tag{7.1}$$

tetszőleges statisztika becslése esetén a helyzet nem ilyen jó: viszonylag ritka, hogy adható a fentihez hasonló képlet a keresett mennyiségekre, illetve az ilyenkor szükséges statisztikai-valószínűségszámítási levezetésbe fektetendő munka is jelentős lehet.

Az ebben a fejezetben bemutatott *bootstrap* módszerek a fenti problémakörre próbálnak egy általános megoldás-sémát adni: az aktuális vizsgálat során elvégzendő statisztikai becsléseket a rendelkezésre álló minta újramintavételezésével próbálják elvégezni.

A fejezet következő részeiben tehát a bootstrap-módszerek részleteit tekintjük át, a következő felosztásban: a 7.1 részben elhelyezzük a bootstrap-módszert annak rokon módszertanai között; a 7.2 és 7.3 részekben áttekintjük a bootstrap alapjait, illetve néhány haladóbb aspektusát; a 7.4 részben a becslések konfidenciájával kapcsolatos eredményeket vizsgáljuk; míg a 7.5 részben a hipotézistesztelést, azon belül is a permutációs teszteket és azok bootstrap változatát ismertetjük.

A bootstrap-módszerek egy részletes tárgyalása iránt érdeklődőknek a [40] könyv ajánlható.

7.1. Ensemble-módszerek áttekintése

A fentiekben láttuk, hogy a bootstrap-módszer legfőbb jellegzetessége a minta felhasználásának módja volt: a segítségével további mintákat állítottunk elő, amelyek együttes használatával végeztünk el bizonyos következtetéseket; ezt tekinthetjük úgy, mintha a x^{*i} bootstrap-minták alapján külön-külön végeztünk volna valamilyen modelltanulási eljárást, amelyek alapján aztán egy közös eredő eredményt alkottunk volna.

Általánosságban azokat a módszereket, amelyek nem egyetlen, hanem több modell alapján működnek, *ensemble*-módszereknek nevezzük. Ezek általános felépítése a következő: (1) adott a tanuláshoz használt megfigyelések (minták) halmaza; (2) ezek alapján elvégezzük modellek egy halmazának tanítását; végül (3) a különböző modellpéldányok által szolgáltatott eredményeket egy valamilyen aggregáló módszer segítségével összekombináljuk.

A következő felsorolásban áttekintjük az ensemble-módszerek közül a legjelentősebb alosztályokat, illetve elhelyezzük ezek között a bootstrap-módszert.

Bayes-optimális osztályozó. A legáltalánosabb, és ezáltal optimális eredményt szolgáltató ensemble-módszer: a Bayes-tételnek megfelelően a lehetséges modellpéldányokhoz a megfigyelési minták alapján számított *a posteriori* valószínűségeket rendeli; a teljes ensemble válaszának meghatározásában minden modell ezzel a súllyal szerepel.

Valós problémák esetén praktikusan nem alkalmazható a teljes modelltér kimerítő bejárásának lehetetlensége miatt.

Bayesi modellátlagolás. Az előző módszer közelítése: a modelltér kimerítő bejárása helyett annak csak egy Monte-Carlo mintavételezése történik meg; a teljes ensemble válasza pedig hasonló súlyozással kerül kiszámításra.

Bayesi modellkombináció. A bayesi modellátlagolással szemben itt nem közvetlenül a modellek terét, hanem a modellterek (a modelltér feletti lehetséges súlyozások) terét mintavételezzük, majd ezek alapján számítjuk az ensemble kimenetét.

Boosting. Az eljárás során az alkalmazott modellek halmazát folyamatosan bővítjük úgy, hogy a korábbiakhoz az azok által rosszul osztályozott mintához nagyobb súlyt rendelő tanítással létrehozott új példányt adunk. Bár ez a módszer gyors eredményekkel kecsegtet, jellegéből adódóan hajlamos a túlilleszkedésre is.

Bucket of models. Ebben az esetben az ensemble kimenetét mindig egyetlen modellpéldány fogja szolgáltatni: mindig a tanulási fázis során legjobban teljesítő modellt fogjuk az adott problémán használni. Másként fogalmazva, maga a modellkiválasztó algoritmus is részt vesz a tanulásban: azt tanulja, hogy az adott problémát melyik modell képes a legjobban megoldani.

A fentiekből látszik, hogy ennek a módszernek akkor van értelme, ha az ensemble-t több problémán is alkalmazni fogjuk, ellenkező esetben az ensemble kimenete pusztán a legjobban teljesítő egyetlen modellpéldány kimenete lesz.

Stacking. Ebben az esetben a tanítás két fázisból áll: (1) először az egyes modelleket tanítjuk, majd (2) a modellek kimenetét összekombináló algoritmus tanítása következik. Elméletben itt állhat bármilyen modell, a leggyakrabban azonban *logisztikus regressziós* modell végzi a modellkimenetek kombinációját.

Bootstrap. Mint azt már a bevezetőben láttuk, a *bootstrap* során az egyes modelleket az eredeti minta (visszatevéses) újramintavételezésével nyert bootstrap-minták alapján nyerjük, majd az egyes kimeneteket egyenlő súllyal kombináljuk.

7.2. A bootstrap alapjai

Mint azt a bevezetőben láthattuk, a bootstrap módszerek célja, hogy az általuk vizsgált eloszlások valamilyen tulajdonságát megbecsüljék a P eloszlásból származó $\mathbf{x}$ minta alapján. Az ilyen mintákhoz kapcsolódó fontos fogalom az *empirikus eloszlásfüggvény* ($\hat{P}$), amely az $\mathbf{x}$ minta elemei feletti egyenletes eloszlás.

A „plug-in" elv. A P eloszlás θ paraméterének ($\theta = s(P)$) *plug-in* becslése a $\hat{\theta} = s(\hat{P})$ mennyiség. Ez a $\hat{\theta}$ becslés pedig a lehető legjobb, amely elérhető, feltéve, hogy a vizsgált P eloszlásról az $\mathbf{x} \sim P$ mintán kívül semmilyen más információ nem áll rendelkezésre.

Standard hiba becslése. A kiindulási szituáció tehát a következő: a $\theta = s(P)$ paramétert becsüljük a $\hat{\theta} = s(\mathbf{x})$ formában, ahol $\mathbf{x} \sim P$. $\hat{\theta}$-val kapcsolatban az alapvető kérdés, hogy mennyire jó becslése ez θ-nak. Erre a legkézenfekvőbb módszer a *standard hiba* becslése, amely a bootstrap-módszerrel a következőképpen történhet:

(1) Az eredeti $\mathbf{x} = x_1, \ldots x_n$ mintából állítsunk elő B darab x_i^* bootstrap mintát, ahol minden x^{*i} mérete egy adott n szám, minden egyes eleme pedig az $\mathbf{x}$ feletti egyenletes eloszlásból származik (vagyis ugyanaz az elem többször is előfordulhat egy bootstrap-mintán belül is).

(2) A vizsgált $s(.)$ statisztikát (a fenti példában ez a várható érték becslése, az átlag) számítsuk ki az összes x^{*i}-re: $s(x^{*1}, \ldots s(x^{*B}))$.

(3) A standard hiba bootstrap-becslése a fentiek alapján a következő:

$$\hat{\mathrm{se}}_{boot} = \sqrt{\sum_{i=1}^{B} \frac{(s(x^{*i}) - \overline{s})^2}{B-1}}, \tag{7.2}$$

(ahol $\overline{s} = \sum_{i=1}^{B} s(x^{*i})/B$), vagyis az $s(x^{*i})$ bootstrap-replikák szórása.

Ha B tart végtelenhez, akkor a fenti becslés tart $\hat{F}$ alapján történő becsléshez:

$$\lim_{B\to\infty} \widehat{\mathrm{se}}_B = \mathrm{se}_{\hat{P}}, \tag{7.3}$$

amelynél a plug-in elv szerint nem lehet jobbat elérni, ha $\mathbf{x}$-en kívül nincs más információnk P-ről.

A fentiek alapján már érthető a módszer elnevezése is, *bootstrap*, vagyis *bocskorszíj* Münchhausen báró egyik ismert történetére utal: ahogyan ő egy tó fenekéről a saját bocskor(csizma)szíjánál fogva húzta fel magát, úgy mi is az egyetlen valódi minta alapján becsüljük annak tulajdonságait.

A fejezet további részeiben a következő jelölési konvenciót követjük: nagybetűk jelölik a valószínűségi változók eloszlását (P, Q), félkövér kisbetűk a belőlük származó mintavektorokat $(\mathbf{x})$, valamilyen (paraméter)érték becsült értékét a $\widehat{\cdot}$ szimbólum jelöli (pl. $\widehat{\theta}$), a $\cdot^*$ szimbólum (potenciálisan kiegészítve egy indexszel) pedig a vonatkozó bootstrap-mintákat (pl. $\mathbf{x}^*$, vagy $\mathbf{x}^{*i}$).

7.3. További aspektusok

Az előző alfejezetben a bootstrap-módszertan alapvető elemeit tekintettük át, itt most néhány haladóbb témakört tekintünk át.

7.3.1. Adatmodell

Az eddigiekben feltételeztük, hogy az $\mathbf{x}$ statisztikai minta egyetlen P eloszlásból származik, elemei pedig egymástól függetlenek, ez az ún. *egymintás* probléma.

Kétmintás probléma. Előfordulhat, hogy a felhasznált minta $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ alakú, ahol a minta két része két különálló eloszlásból származik: $\mathbf{y} \sim Q$, $\mathbf{z} \sim R$ (ilyen lehet például egy klinikai tesztben az esetek és kontrollok csoportja). Ebben az esetben természetesen nem megfelelő a $P = (Q, R)$ együttes eloszlást x-ből egyenletes eloszlással vett mintákkal replikálni, hanem a két csoport bootstrap-mintáit külön-külön előállítva kell a teljes mintákat előállítani.

Idősorok kezelése. Ha a vizsgált adatok egy idősort alkotnak (vagyis az aktuális minták a korábbiaktól függnek valahogyan), a független mintavételezés nem lehet megfelelő. Ilyenkor az egyik lehetőség az *autoregressziós* modell használata, vagyis annak feltételezése, hogy az idősor elemei felírhatók a következő formában:

$$z_t = \left(\sum_{i=1}^{m} \beta_i z_{t-i} \right) + \epsilon_t, \tag{7.4}$$

ahol z_t-k az eredeti x_t-kből úgy állnak elő, hogy azokból kivonjuk azok μ várható értékét: $z_t = x_t - \mu$ (vagyis z x-nek annyival eltolt változata, hogy a várható értéke 0 legyen); β_i-k az autoregressziós együtthatók, ϵ_t-k pedig az egymástól független (elemenkénti) zajparaméterek 0-nak feltételezett várható értékkel.

A β_i-k bootstrap-becslése (feltételezve, hogy a β vektor hosszát ismerjük) a következők szerint haladhat:

(1) Az eredeti minta alapján például a *legkisebb négyzetek módszerével* becsüljük meg β értékét: $\hat{\beta}$.

(2) $\hat{\beta}$ és a 7.4 egyenlet alapján meg tudjuk becsülni a zajparaméterek értékeit: $\hat{\epsilon}_t$.

(3) Az idősor z^* bootstrap replikáit ezek után már le tudjuk generálni a 7.4 egyenlet rekurzív felhasználásával, ϵ_t^*-ot $\hat{\epsilon}$-ból mintavételezve.

(4) A generált bootstrap idősorokból a megfelelő β^* értékek már előállíthatók.

A fentinél az egymintás megközelítéshez jobban hasonlító módszer a *mozgóblokkos* bootstrap alkalmazása. Ebben az esetben a bootstrap replikákat az eredeti idősornak (egy előre meghatározott) n hosszú, véletlenszerűen kiválasztott kezdéspontú szakaszainak egymás után fűzésével állítjuk elő.

7.4. Konfidencia becslések

A fejezet korábbi részeiben eddig pontbecslésekkel foglalkoztunk, azaz a vizsgált mennyiségre egyetlen értéket próbáltunk adni; statisztikai vizsgálatok során azonban egy jó közelítést adó pontbecslés mellett ugyanolyan fontosak a megbízhatóságra (*konfidenciára*) vonatkozó kijelentések, amelyek egy adott halmazba/intervallumba adott valószínűséggel való tartozást vizsgálnak, mint például:

> Az $s(.)$ statisztika az értékét $P = 90\%$ valószínűséggel a $[c_{lo}, c_{hi}]$ intervallumban veszi fel.

A fenti kijelentésben a P valószínűséget a konfidencia *szintjének*, a $[c_{lo}, c_{hi}]$ intervallumot pedig a P valószínűségű *konfidencia intervallumnak* nevezzük.

Ha a szokásos módon $\hat{\theta} = s(\hat{P})$ a $\theta = s(P)$ paraméter becslése, akkor (kellően általános feltételek mellett) $\hat{\theta}$ megközelítőleg a $\mathcal{N}(\theta, \hat{\text{se}}^2)$ normális eloszlást fogja felvenni, vagyis:

$$\frac{\hat{\theta} - \theta}{\widehat{\text{se}}} \dot{\sim} \mathcal{N}(0, 1). \tag{7.5}$$

A fentiek alapján tehát a pontbecslések konfidencia intervallumai könnyen meghatározhatók a normális eloszlás percentilisei alapján: ha például a $P = 90\%$-os konfidencia intervallumra vagyunk kíváncsiak, akkor ahhoz a $\frac{P}{2} = \alpha$ és az $1 - \frac{P}{2} = 1 - \alpha$ percentilisek értékeit kell felhasználnunk, amelyek így a $[\hat{\theta} - \mathcal{N}^{(1-\alpha)}\widehat{\text{se}}, \hat{\theta} - \mathcal{N}^{(\alpha)}\widehat{\text{se}}]$ intervallumot adják (ahol $P^{(\alpha)}$ jelöli a P eloszlás α percentilisét).

Mivel a fenti becslés alapjául a normális eloszláshoz való tartás szolgál, az csak akkor helytálló, hogyha ez a konvergencia valóban megvalósul legalább közelítő jelleggel, vagyis a rendelkezésre álló minták száma kellően nagy. Kisebb mintaszámok (kb. $n = 100$ alatt) jobb közelítést ad a normális eloszlás helyett az $n - 1$ szabadsági fokú *Student-t* eloszlás.

Bootstrap-t intervallum. A fentiek alapján meg tudjuk konstruálni a t-eloszlás bootstrap változatát, amely a szokásos módon valószínűségelméleti háttérmegfontolások nélkül, kizárólag a megfigyelt minta felhasználásával születik. A fő lépések a következők.

Az $x^{*1}, \cdots x^{*B}$ bootstrap minták alapján kiszámítjuk a

$$Z^*(i) = \frac{\hat{\theta^*}(i) - \hat{\theta}}{\widehat{\text{se}}^*(i)} \tag{7.6}$$

elemeket, ahol $\hat{\theta}^*(i) = s(x^{*i})$ a θ paraméter becslése az i-edik bootstrap minta alapján, $\widehat{\text{se}}^*(i)$ pedig az ugyanerre a mintára vonatkozó standard hiba bootstrap-becslése. Ezek alapján a $Z^*(.)$ eloszlás α percentilisének becslése a következő lesz:

$$\hat{t}^{(\alpha)} : \frac{|Z^*(i) \leq \hat{t}^{(\alpha)}|}{B} = \alpha. \tag{7.7}$$

Így pedig a bootstrap-t alapján adódó konfidencia intervallum $[\hat{\theta} - \hat{t}^{(1-\alpha)}\widehat{\text{se}}, \hat{\theta} - \hat{t}^{(\alpha)}\widehat{\text{se}}]$ lesz.

Bár a bootstrap-t szépen alkalmazható az egyszerűbb esetekben, egy súlyos hiányossága, hogy nem érzéketlen a becsült mennyiségekre alkalmazott transzformációkra, vagyis, ha θ helyett egy $\phi = t(\theta)$ paraméterre végezzük el a becslést, majd a kapott intervallumhatárokra alkalmazzuk a t^{-1} transzformációt, akkor nem kapjuk vissza a közvetlenül θ becslésére kapott értékeket. Ezek szerint minden paraméter előnyös lenne megkeresni azt a transzformációt, amely alkalmazásával a legszűkebb intervallumot kapjuk.

Mivel az ilyen transzformáció megtalálása/kiszámítása nem mindig triviális feladat, a következőben bemutatunk egy módszert, amely kiküszöböli a bootstrap-t e hiányosságát.

Bootstrap-percentilisek. Az előzőekben láttuk, hogy ha $\hat{\theta}$ a θ paraméter becslése, $\widehat{se}$ pedig a becslés standard hibája, akkor a $P = 1-2\alpha$ konfidenciaszinthez tartozó konfidencia intervallum $[\hat{\theta}^{*(\alpha)}, \hat{\theta}^{*(1-\alpha)}]$ lesz, ahol $\hat{\theta}^* \sim \mathcal{N}(\hat{\theta}, \widehat{\text{se}}^2)$.

Ez már sugallja, hogy hogyan fog működni a most tárgyalt algoritmus: (1) állítsuk elő a $\hat{\theta}$ becslésre vonatkozó $\hat{\theta}^*$ bootstrap értékeket, (2) becsüljük a percentiliseket ezek alapján.

A kiindulási ötlethez visszatérve, a központi határeloszlás tétele alapján tudható, hogy a $\hat{\theta}^*$ bootstrap-becslés eloszlása tart a normálishoz, ha az eredeti n mintaméretre: $n \to \infty$. Nagy mintaméretre tehát a bootstrap percentilisek alapján adódó intervallum meg fog egyezni a normálissal, és az is belátható, hogy kis mintaszámok esetén jobban is fog viselkedni. Ezeken felül ez a módszer érzéketlen a θ paraméter transzformációira is, vagyis bármely $\hat{\phi} = t(\hat{\theta})$ transzformált becslésre a $\hat{theta}$-hoz tartozó intervallum $[t^{-1}(\hat{\phi}^{*(\alpha)}), t^{-1}(\hat{\phi}^{*(1-\alpha)})]$ lesz.

7.5. Permutációs teszt és hipotézistesztelés

Az alapprobléma a következő, adott két statisztikai minta:

$$P \rightarrow \mathbf{y} = (y_1, \ldots y_m) \tag{7.8}$$

$$Q \rightarrow \mathbf{z} = (z_1, \ldots z_n), \tag{7.9}$$

kérdés, hogy a két eloszlás, amelyekből származnak, megegyezik-e, azaz a vizsgált nullhipotézis: $H_0 : P = Q$.

Bármiféle hipotézistesztelés esetén a kiindulási pont egy $\hat{\theta}$ teszt-statisztika konstruálása, amely tipikusan akkor vesz fel nagy értékeket, amikor H_0 hamis (a fenti példában például az átlagok különbsége $\hat{\theta} = \overline{y} - \overline{z}$ egy megfelelő választás lehet).

$\hat{\theta}$ megfigyelése után a teszt által szolgáltatott *szignifikancia-szint* (más néven *p-érték*) a $P_{H_0}(\hat{\theta}^* \geq \hat{\theta})$ valószínűség lesz, vagyis annak a valószínűsége, hogy H_0 fennállása esetén a valóban megfigyelt $\hat{\theta}$ értéknél nagyobbat[1] kaphattunk volna.

Permutációs teszt. Általános esetben $\hat{\theta}$ eloszlásának meghatározása (ami alapján dönthetnénk H_0 elvetéséről vagy elfogadásáról) igen bonyolult lehet; ezt a problémát kezeli a *permutációs teszt* módszere, amely különösebb előfeltevések nélkül képes a fenti H_0-lal kapcsolatban eredményeket szolgáltatni.

A permutációs teszt a következőképpen halad:

(1) Az eredeti $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ mintából előállítjuk $\mathbf{v}$-t, amely $\mathbf{x}$ sorrendezett értékeit tartalmazza (ezen belül tehát az eredeti $\mathbf{y}$ és $\mathbf{z}$ értékei tipikusan vegyes sorrendben szerepelnek), valamint $\mathbf{g}$-t, amely azt tartalmazza, hogy $\mathbf{v}$-n belül melyik elem hova tartozott eredetileg.

(2) Ekkor $\mathbf{g}$ összesen $\binom{(m+n)}{n}$ lehetséges értéket vehet fel, amelyek felett a *permutációs lemma* szerint H_0 fennállása esetén egyenletes az eloszlása.

(3) Sorsoljunk B darab független $\mathbf{g}^*(1), \ldots \mathbf{g}^*(B)$ vektort a lehetséges értékek halmazából.

(4) Számítsuk ki a vonatkozó $\hat{\theta}^*(i)$ statisztikát a fenti mintákra.

(5) A p-értékre adott becslés ezek után a következő lesz:

$$\widehat{\text{p–value}} = \frac{|\hat{\theta}^*(i) \geq \hat{\theta}|}{B}. \tag{7.10}$$

A fenti módszer és a bootstrap által szolgáltatott konfidenciatesztek közti hasonlóság szembeötlő: itt is egy újramintavételezett mintán, a percentilisekhez hasonló becslést végzünk.

Ennek megfelelően a permutációs teszt bootstrap változata a következő:

(1) Sorsoljunk B darab $m + n$ méretű mintát $\mathbf{x} = (\mathbf{y}, \mathbf{z})$-ből (visszatevéssel); tekintsük úgy, hogy az első m minta $\mathbf{y}$-ból, az utolsó n pedig $\mathbf{z}$-ből származik.

(2,3) Értékeljük ki a $\hat{\theta}^*(i)$ statisztikát a bootstrap mintákra, majd becsüljük a p-értéket a már ismert képlettel: $\widehat{\text{p–value}} = \frac{|\hat{\theta}^*(i) \geq \hat{\theta}|}{B}$.

[1] Természetesen $\hat{\theta}$-t nem feltétlenül kell úgy megválasztani, hogy a nagyobb értékek jelentsék H_0 elvetését, mi azonban a továbbiakban is ezt a konvenciót követjük.

8. fejezet

Kernel technikák az intelligens adatelemzésben

8.1. Bevezetés

A számítás- és méréstechnika fejlődése a XX. század végén tovább gyorsult. A gyorsuló fejlődés egyre nagyobb adathalmazokat eredményez, amelyek rengeteg információt tartalmaznak újabb kihívások elé állítva az adatelemzőket, statisztikusokat. Több nagy kutatási terület is használ vagy épít ilyen nagy adatbázisokat, például fizikában a CERN részecskegyorsítóban keletkező adatok, biológiában az 1000 genom projekt, de akár egy teljes genom szekvenálása során is rengeteg adat keletkezik. Ezekről az adatforrásokról sokszor rendelkezünk valamilyen plusz információval, például az adat struktúrájáról. Rengeteg adatbázis reprezentálható gráfokként. Ilyen tipikus gráf reprezentáció lehet egy fehérje–fehérje interakciós hálózat, ahol fehérjék a csomópontok és két fehérje között akkor van él, ha a két fehérje valamilyen interakcióban van egymással. Ehhez hasonlóak a génregularizációs hálók, csak itt gének, alkotják a csomópontokat az éleket pedig a szabályozó kapcsolatok. A bioinformatikától távol is találunk ilyen adatbázisokat, elég a weblapokra gondolni, amik között a kapcsolatot az egymásra mutató linkek teremtik meg, vagy a ismeretségi adatbázisok (facebook, linkedIn) ahol az egyes személyek a csomópontok, és a kölcsönös ismertség adja az éleket.

A könyvben már volt szó kernel módszerekről és support vektor gépekről. Most ezt az irányt folytatva bemutatjuk, hogy a különböző kernel módszerekhez, hogyan lehet előzetes információt (priort) felhasználva kernelt gyártani. A fenti területeknél maradva, ilyen információ lehet akár egy transzformáció invariancia, akár egy bizonyos típusú eloszlás, melynek megfelelő megoldást keresünk, vagy a populáció leírása ahonnan az adatunk származik. Az első részben röviden bemutatjuk a leggyakrabban használt kernel függvényeket, és azt, hogy létező kernelek kombinációjából, hogyan lehet újabb kerneleket építeni. Ezután bemutatunk néhány módszert, melyek segítségével a prior ismereteinket használhatjuk fel a tanításban, vagy a tanító halmaz kiegészítésével, vagy akár az ismeretek kernelbe ágyazásával. Ezt követően megmutatjuk, hogyan lehet gráf kernelt építeni kihasználva a gráf ismert szerkezetét. Végül a teljesség igénye nélkül mutatunk példát né-

8.1. táblázat. Legelterjedtebben használt kernel függvények [4]

Lineáris	$K(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{x}^T \boldsymbol{x'}$
Polinomiális	$K(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x}^T \boldsymbol{x'} + 1)^d$
Gauss	$K(\boldsymbol{x}, \boldsymbol{x'}) = e^{\frac{-\|\boldsymbol{x}-\boldsymbol{x'}\|^2}{\sigma^2}}$
Cosinus	$K(\boldsymbol{x}, \boldsymbol{x'}) = \cos(\angle(\boldsymbol{x}, \boldsymbol{x'}))$
Tangens hiperbolikusz	$K(\boldsymbol{x}, \boldsymbol{x'}) = \tanh(k\boldsymbol{x}^T \boldsymbol{x'} + \theta)$

hány, elsősorban bioinformatikában használt adatbázisra, ahol tipikusan lehet alkalmazni a fejezetben leírt módszereket. A téma iránt mélyebben érdeklődőknek ajánljuk a [78] cikk és a kernel módszerek esetén alapműnek minősülő [99] könyv olvasását.

8.2. Kernelek konstrukciója

Ebben a részben először bemutatjuk a leggyakrabban használt kerneleket, és olyan műveleteket, melyek megtartják a kernelek alaptulajdonságait. A fejezet elsősorban a [99] alapművet követi, a téma iránt jobban érdeklődők számára a [99] 13. fejezetét ajánljuk további olvasásra.

A különböző kernel módszerek mögött ugyanaz a gondolat áll: nevezetesen kifejezni az Ω tér egyes pontjai közötti hasonlóságot a $K : \Omega \times \Omega \to \mathbb{R}$ kernel függvénnyel. Amennyiben nem lineárisan szeparálható a probléma a kernel konstrukció magában foglalja a jellemző térbe való Φ leképezést a H_k Hilbert térbe: $\Phi : \Omega \to H_k$, ahol a kernel egy skalár szorzatként előáll.[73]

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \tag{8.1}$$

8.2.1. Leggyakrabban használt kernel függvények

Nem minden feladat esetén szükséges probléma specifikus kernelt gyártani. Ebben a részben bemutatunk pár elterjedten használt kernel függvényt, melyek kielégítik a Mercer–feltételeket, tehát pozitív szemidefinit kernel mátrixot adnak. Az 8.1 táblázatban d, σ, k és θ konstansok. A tangens hiperbolikusz kernel függvény esetén megfelelően kell megválasztani a két konstanst, mivel nem minden paraméterkombináció eredményez kernel függvényt.

8.2.2. Műveletek kernelekkel

Ismert, hogy a kernelek tere konvex kúpot alkot és zárt a pontonkénti konvergenciára nézve [99]. Így, ha $k_1(x, x')$ és $k_2(x, x')$ kernelek, továbbá $\alpha_1, \alpha_2 \geq 0$, akkor

$$k(x, x') = \alpha_1 \cdot k_1(x, x') + \alpha_2 \cdot k_2(x, x') \tag{8.2}$$

is kernelek. Tehát kernelek nemnegatív lineáris kombinációja is kernel. Emellett, ha $k_1, k_2, ..., k_n$ kernelek és

$$k(x, x') := \lim_{n\to\infty} k_n(x, x') \tag{8.3}$$

létezik $\forall x, x'$ esetén, akkor k kernel. Továbbá kernelek szorzata is kernel:

$$(k_1 k_2)(x, x') := k_1(x, x') \cdot k_2(x, x') \tag{8.4}$$

Ebből következik, hogy az 8.2.1. állítás alapján is számíthatunk újabb kerneleket.

8.2.1. Állítás. *Ha $k_1(x_1, x_1')$ és $k_2(x_2, x_2')$ a $X_1 \times X_1$ és $X_2 \times X_2$ tér felett definiált kernelek, $(x_1, x_1' \in X_1$ és $x_2, x_2' \in X_2)$, akkor külső szorzatuk,*

$$(k_1 \otimes k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1')k_2(x_2, x_2') \tag{8.5}$$

is kernel $(X_1 \times X_2) \times (X_1 \times X_2)$ tér felett.

Az említett műveletek alkalmazásával könnyen lehet kernelekből újabb kerneleket előállítani, és az 8.2.1. állításból jól látható, hogy adott esetben, ha egy adott feladatot fel lehet bontani kettő vagy több részfeladatra, melyek megoldásához rendelkezünk már kernelekkel, akkor ezek külső szorzatából elő lehet állítani egy közös kernelt, amivel a teljes feladat megoldását megadhatjuk.

8.3. Prior információ hozzáadása

Adatelemzés során sokszor többlet információkkal rendelkezhetünk az adathalmazról. A [99] definícióját használva prior információnak tekintünk a tanító halmazban nem szereplő minden információt. Ilyenkor több lehetőségünk is van arra, hogy felhasználjuk ezeket az információkat. Kiegészíthetjük a tanító halmazt újabb pontokkal, a kernel függvénybe ágyazhatjuk a prior ismereteket vagy az optimalizációs függvényt módosíthatjuk. A jelen fejezet csak az első kettőt tárgyalja, az optimalizációt változtató módszerekről az [78] cikkben található egy jó összefoglaló. Az alábbiakban csak a két-osztályos esetekre írjuk le az egyes módszereket, ám mindegyik algoritmus könnyen kiterjeszthető több-osztályos feladatokra. A 8.3 részhez és besoroláshoz a [78] cikket vettük alapul.

8.3.1. Tanító halmaz bővítése

Sokszor a tanító halmaz bővítése a legkönnyebben járható út. Ilyenkor a tanító halmazhoz adunk újabb minta pontokat, amelyek a prior ismereteinket tükrözik. Két alapvető módszert mutatunk be a virtuális minták hozzáadását és a tanító pontok súlyozását.

Virtuális minta pontok

Virtuális minta pontokat többféleképpen generálhatunk. Általában valamilyen transzformáció invariancia esetén használjuk ezt a módszert. Ilyenkor a tanító pontjaink egyszerűen előállnak a transzformáció alkalmazásával.

$$f(x) = f(Tx) \tag{8.6}$$

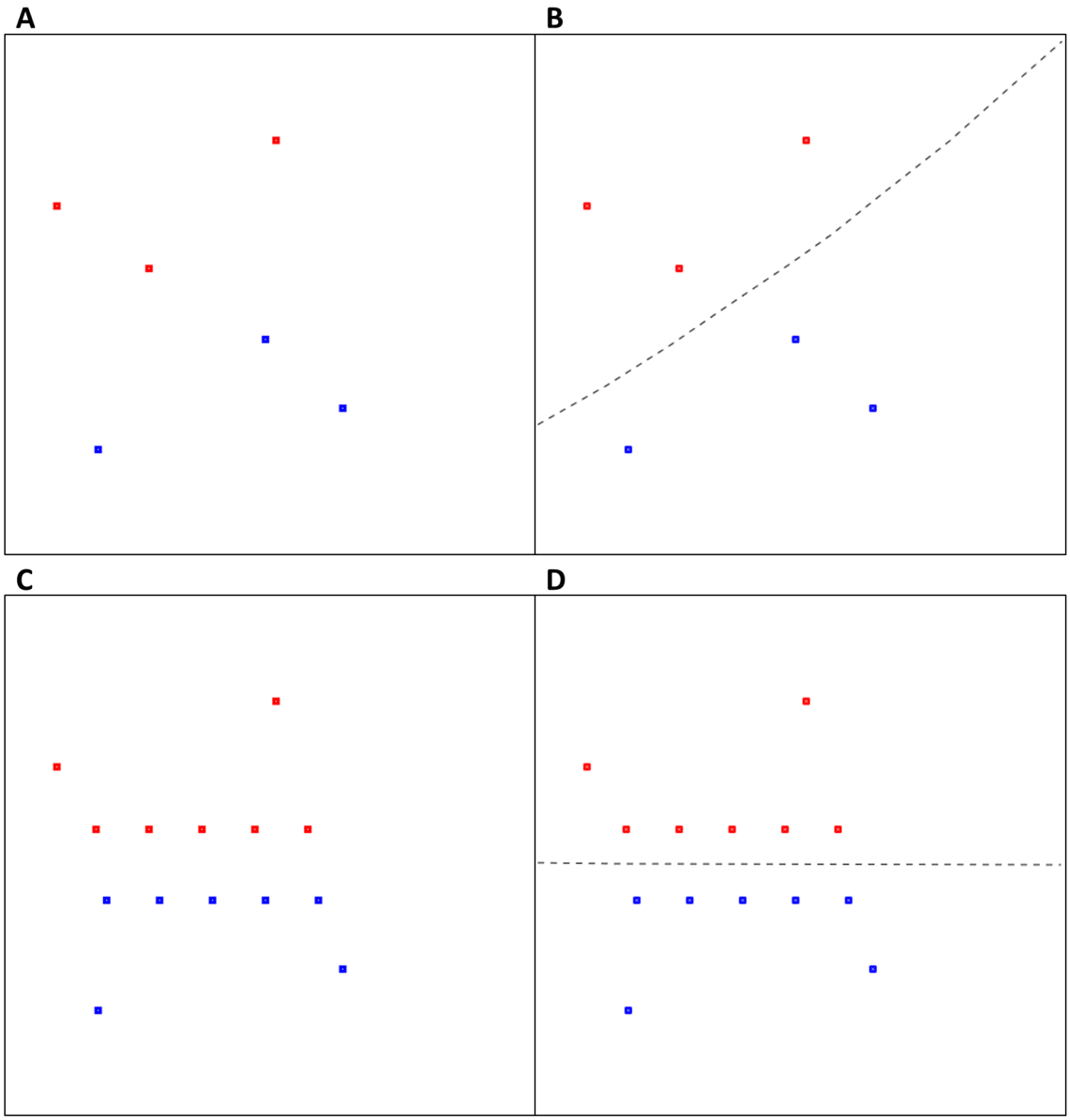
ahol $f(x)$ az osztályozó függvény és T az invariáns transzformáció. Az 8.1 ábra mutat erre egy példát. Kezdetben viszonylag kevés tanítópont áll rendelkezésre. Ezek alapján kell az osztályozási feladatot megtanulni. Továbbá ismert, hogy az adatok invariánsak az eltolásra egy adott irányban, tegyük fel, hogy jobbra vagy balra tolva az egyes tanítópontokat nem változtatja meg az így keletkező új tanítópont osztályát. Ha csak a tanító halmaz alapján tanítjuk az SVM-et az 8.1/B ábrán látható szeparálási görbét kapjuk. Jól láthatóan akármelyik piros pontot toljuk el jobbra, egy idő után sérti a prior információnk alapján feltett osztályinvariáns eltolást. Ezért kiegészíthetjük az tanító halmazt a meglévő tanítópontokat megfelelően eltolva jobbra és balra. A tanulás után a szeparáló görbe teljesíti a prior elvárásainkat.

A Virtuális Support Vektorok (VSV) módszere két lépésben tanít egy support vektor gépet (SVM). Első lépésben a meglévő tanító halmazon végez tanulást, majd az így keletkezett support vektorokat felhasználva generál virtuális mintákat, azaz virtuális support vektorokat. Ugyanis SVM tanulás során a keletkező support vektorok tartalmaznak minden szükséges információt az adatról.

Ezeket a módszereket alkalmazzák képfeldolgozási feladatokban például karakter felismerésre. Könnyen belátható, hogy egy adott karaktert eltolva, vagy elforgatva még mindig ugyanazt a karaktert kapjuk, így ugyanabba az osztályba kell sorolnia az osztályozónak. Ilyenkor a már meglévő osztályozott karakterekből lehet több új minta pontot legyártani.

Tanító pontok súlyozása

Az osztályinvariáns transzformációk mellett más jellegű prior információk megjelenítésére alkalmas a minta pontok súlyozása. Ha az egyik osztály felülreprezentált a tanító halmazban, vagy rendelkezésre állnak minőségi mutatók az egyes minta pontokhoz, akkor alkalmazhatjuk ezt a módszert. Ilyenkor különböző költséget rendelünk az egyes osztályokhoz a soft-margin SVM-et tanítva. A kisebb osztály esetén nagyobb költséget megadva, így jobban büntetve kisebb elemszámú osztály esetén a rossz osztályozást biztosíthatjuk, hogy a kevés minta pontot pontosan osztályozzuk. Ehhez hasonlóan, ha rendelkezünk valamilyen minőségi mutatóval a minta pontokról, akkor minden ponthoz meghatározhatunk külön-külön költséget. Végül, ha adott nem osztályba sorolt minta pontokról sejtjük, hogy melyik osztályba tartoznak, akkor adhatunk hozzájuk úgynevezett pszeudo-címkét. Ebben az esetben a minta pontokhoz rendelt költségben jelenítjük meg, hogy mennyire vagyunk biztosak az adott információban.



*8.1. ábra. SVM tanítása prior invariancia ismeretében a tanító halmaz kiegészítésével. **A** ábrán látható az eredeti tanító halmaz. **B** az eredeti tanító halmazon való tanítás eredményét mutatja. **C** a kiegészített adathalmazt adja meg, végül **D** ábrán látható a kiegészítést követő tanítás eredménye. A tanításhoz a LibSVM SVMtoy [20] csomagját használtuk.*

8.3.2. Prior információ kernelbe ágyazása

A tanító halmaz további bővítése mellett közvetlenül a kernelbe is ágyazhatjuk a priort. Ilyenkor a kernel mátrixot módosítjuk úgy, hogy tükrözze a prior információt. A jitter, tangens és Haar-integrációs kernelek a virtuális mintákhoz hasonlóan transzformáció inva-

riánssá teszik a kernelt. A tangens kernelek a bemeneti térben definiálnak egy új távolságot és ezt használják fel kernel számításra, míg a Haar-integrációs kernelek transzformáció halmazok invarianciájának kernelbe ültetésére alkalmasak. A jitter kerneleket részletesebben tárgyaljuk. A fentiek mellett a permutáció invarianciát kernelbe építő, halmazok közötti kernelek módszert, és a tanító halmaz kiegyensúlyozatlanságát figyelembe vevő tudás alapú kernel gyártást mutatjuk be röviden.

Jitter kernelek

A VSV módszerhez hasonlóan transzformáció invariancia esetén lehet a módszert alkalmazni. A kernelt úgy módosítjuk, hogy a kernel mátrix egyes elemeit cseréljük. Vesszük x_i-t és az összes, osztályra nézve invariáns transzformáltját, ezek közül kiválasztjuk azt (x_q) amelyik minimalizálja

$$\sqrt{k_{qq} - 2k_{qj} + k_{jj}} \tag{8.7}$$

kifejezést. Végül $k_{ij} = k_{qj}$ cserét hajtjuk végre.

Kernelek halmazok között

Ha az egyes tanítópontokat különböző halmazokba sorolhatjuk, akkor ezek között a halmazok között is lehet kernel definiálni, így lehetővé az invarianciát az egyes halmazokon belül a permutációra nézve. A kernelt a Bhattacharyya együttható alapján építjük fel.

$$k(\chi, \chi') = k(p, p') = \int \sqrt{p(x)}\sqrt{p'(x)}dx \tag{8.8}$$

ahol χ és χ' a vektorokból épített két halmaz, és p és p' ezeket a halmazokra illesztett eloszlások x pedig egy eleme a halmaznak. Az p, p' eloszlások halmazokra való illesztése teszi lehetővé a permutáció invarianciát. Olyan esetekben alkalmazzák a módszert, amikor az osztályozást nem befolyásolhatja az egyes halmazokban a vektorok sorrendje. Például $[x, y, \gamma]$ koordináta, szürkeárnyalat értékekkel írunk le egy képet. Bioinformatikában haplotípusba, mint osztályba soroláskor használhatjuk ezt a módszert.

Tudás alapú kernel konstrukció

Amellett, hogy az adatunk invariáns lehet transzformációkra, sokszor olyan priorral rendelkezünk, ami az osztályokról ad információt. Ismert lehet az egyes osztályok kiegyensúlyozatlansága, amikor az egyik osztályt nagyjából lefedik a rendelkezésre álló minta pontok, míg a másik osztályban a minta pontok csak egy kisebb részét reprezentálják a teljes halmaznak. Ilyen feladat lehet az arc felismerés, amikor a kis létszámú pozitív osztály megfelelően reprezentált, de nem lehet a negatív osztályba elegendő tanító pontot találni. Hasonlóan genetikában egy génszabályozási háló bővítése is hasonló feladat, hiszen kevés vagy nulla szabályozó elemről állíthatjuk biztosan, hogy nincs szerepe az adott szabályozási hálóban. Ilyenkor alkalmazhatjuk a tudás alapú kernel konstrukciót. A módszer

lényege, hogy a pozitív mintákat egy kisebb pontba húzza össze, míg a negatív mintákat a tér nagyobb részére teríti szét. A kernelt a következőképp definiáljuk:

$$J(k,\Theta) = \frac{tr(S_{np}^{\Phi})}{tr(S_{p}^{\Phi})} \tag{8.9}$$

$$S_{np}^{\Phi} = \sum_{x_i \in D_n} (\Phi(x_i) - m_p^{\Phi})(\Phi(x_i) - m_p^{\Phi})^T \tag{8.10}$$

$$S_{p}^{\Phi} = \sum_{x_i \in D_p} (\Phi(x_i) - m_p^{\Phi})(\Phi(x_i) - m_p^{\Phi})^T \tag{8.11}$$

ahol m_p^{Φ} a pozitív elemek átlaga, $\Phi(x_i)$ a jellemző térbe való leképezés, $tr(S_{np}^{\Phi})$ pedig az adott mátrix nyoma.

8.4. Kernelek gráfokra

A kernelekre vonatkozó megkötések, azaz egy kernel mátrix mindig szimmetrikus, pozitív definit, megnehezítik a nem euklideszi terekben való alkalmazást. Ilyen esetekben sokszor először euklideszi térbe transzformálják a bemeneti teret. Az alábbiakban egy olyan módszert mutatunk be, mellyel gráfokhoz lehet kernelt gyártani, így lehetővé válik gráfok összehasonlítása is, vagy egy adott gráfra kernel készítése.

Egy gráf csomópontok (V) és élek halmaza (E), ahol az élek csomópontok nem sorba rendezett párjai $\{v_i, v_j\} \in E$, ha $v_i \in V$ és $v_j \in V$ egy éllel van összekötve.
Jelölése: $v_i \sim v_j$.

Az úgynevezett diffúziós kerneleket az exponenciális kernelekből vezetjük le. Megmutatjuk, hogy definíció szerint pozitív szemidefinit és szimmetrikus kernelt kapunk. Egy négyzetes H mátrix exponenciálisa a következő:

$$e^{\beta H} = \lim_{n\to\infty} \left(I + \frac{\beta H}{n}\right)^n \tag{8.12}$$

A fenti határérték mindig létezik és ekvivalens az alábbi sorösszeggel.

$$e^{\beta H} = I + H + \frac{1}{2!}H^2 + \frac{1}{3!}H^3 + ... \tag{8.13}$$

Továbbá ismert, hogy minden szimmetrikus mátrix páros hatványa pozitív szemidefinit, és a pozitív szemidefinit mátrixok halmaza konvex kúpot alkot és zárt a pontonkénti konvergenciára nézve. Ebből az állításból következett a 8.2.2 részben, hogy a kernelek is konvex kúpot alkotnak. Tehát H-t szimmetrikusnak választva és n helyett $2n$-et véve, a mátrix exponenciálisa szimmetrikus és pozitív szemidefinit lesz. Így kapunk egy megfelelő kernel jelöltet. Az ebben a formában előállított kernelnek a β szerinti parciális deriváltja:

$$\frac{d}{d\beta}K_\beta = HK_\beta \tag{8.14}$$

A $\beta = 0, K_0 = I$, kezdeti feltétellel, ha H-t úgy választjuk meg, hogy a lokális struktúráját jelenítse meg az Ω térnek, akkor a kernelben automatikusan megjelenik Ω globális struktúrája. Egy gráfra nézve ez azt jelenti, hogy H-t az egyes csomópontok közötti kapcsolatnak megfeleltetve a kernelben a teljes struktúra szerepelni fog. H-t a K kernel generátorának, β-t pedig a sávszélességének nevezzük. Az eddigi általános leírás után áttérünk a fenti módszerrel generált kernel gráfokra történő alkalmazására.

8.4.1. Diffúziós kernelek

Definiáljuk H-t a következőképp:

$$H_{ij} = \begin{cases} 1 & \text{ha i} \sim \text{j} \\ -d_i & \text{ha i} = \text{j} \\ 0 & \text{egyébként} \end{cases} \tag{8.15}$$

ahol d_i az i csomópont foka (i-t érintő élek száma). Amennyiben a gráf súlyozott: legyen w_{ij} a $\{v_i, v_j\} \in E$ él súlya. Ekkor az éleknek vesszük az összegét $H_{ij} = \sum w_{ij}$ és $i \neq j$. Ennek megfelelően kell súlyozni H diagonális elemeit is.

Megmutatható, hogy a diffúziós kernelek vagy hő kernelek természetes választásnak tűnnek gráfok esetén. Megfelelő paraméter választással véletlen bolyongásnak is megfeleltethető a diffúziós kernel.

8.5. Adatbázisok

Az alábbi adatbázisok kiindulópontul szolgálhatnak, hogy milyen problémák esetén lehet jól alkalmazni a leírt kernel technikákat. Szerepel köztük több genetikai információt tároló adatbázis, de akad például képfeldolgozási, osztályozási problémákhoz alap feladatnak tekinthető adat is. Nagy részük publikus, bárki számára elérhető, míg a TRANSFAC például előfizetéshez kötött. A jelen fejezetben nincs lehetőségünk az adott adatbázisokon alkalmazott kernel módszerek és eredmények bemutatására. Az érdeklődők számára kiindulópontul szolgálhat a [99] könyv. Bioinformatikai alkalmazásokban prioritizálásra többen használnak kerneleket egy átfogóbb, ugyanakkor nem teljes leírása ezeknek a módszereknek megtalálható a [112] cikkben.

miRNA adatbázisok

TarBase [117], mirTarBase [66] és a mirBASE [74] egy folyamatosan frissülő microRNS adatbázis. Az adatbázis tartalmazza az érett microRNS szekvenciákat és ezek helyét a genomban, illetve annotációit. Megkereshetjük az egyes microRNS-ek célpontjait. Egy microRNS-nek több célpontja is lehet, míg egy cél gén több microRNS-t is meg tud kötni. Ezekből a kapcsolatokból felépíthetjük a microRNS-ek és target gének kapcsolati hálóját.

TRANSFAC

A TRANSFAC eukarióták transzkripciós faktorait és ezek kötőhelyeit tároló adatbázis. Van publikusan elérhető verziója, ám ez kevesebb információt tartalmaz. Az egyes transzkripciós faktorokat és a kötőhelyeket is kísérletekkel validálják. Kereshetőek benne a transzkripciós faktorok által szabályozott fehérjék, és a TRANSCompel modul tárolja az ismert együttesen – akár szinergetikusan, akár antagonisztikusan – ható faktorokat is. [84]

HGMD

A Human Gene Mutation Database (Emberi Génmutációk Adatbázisa) azokat az emberben megtalálható génmutációkat tárolja, melyeket összefüggésbe hoztak valamilyen betegséggel. Van nyilvános és előfizetéssel elérhető változata, az utóbbi jelentősen több információt tartalmaz.[106]

STRING

A String fehérje interakciókat tartalmazó adatbázis [44]. Kísérletileg támogatott és *in silico* predikciók eredményei is szerepelnek benne. Az egyes kapcsolatokat együttes expressziós mintázatok, genomikai környezet és nagy áteresztő képességű mérések alapján határozzák meg. Több fajról tárolnak adatot.

JASPAR

A JASPAR a TRANSFAC-hoz hasonlóan eukarióták kísérletesen bizonyított transzkripciós faktorait és ezek kötőhelyeit leíró adatbázis, viszont a TRANSFAC-tól eltérően a JASPAR publikus. [14]

US Postal Service

Az amerikai posta által összegyűjtött nagyjából, 12000 kézzel írt számjegyet tartalmaz. Az adat jó minta egy digitális szám felismerő algoritmus tanításához, teszteléséhez. [67]

Jelölések[1]

Felhasznált jelölések

x,$\underline{x}$,$\underline{\underline{x}}$	skalár, (oszlop)vektor vagy halmaz, mátrix
$X, x, p(X)$	véletlen változó X, érték x, valószínűségi tömegfüggvény/sűrűségfüggvény X
$\mathrm{E}_{X,p(X)}[f(X)]$	$f(X)$ várható értéke $p(X)$ szerint
$\mathrm{var}_{p(X)}[f(X)]$	$f(X)$ varianciája $p(X)$ szerint
$I_p(\underline{X}\|\underline{Z}\|\underline{Y})$	$\underline{X}$ és $\underline{Y}$ megfigyelési függetlensége $\underline{Z}$ feltétellel p esetében
$(X \perp\!\!\!\perp Y\|Z)_p$	$I_p(\underline{X}\|\underline{Z}\|\underline{Y})$
$(X \not\perp\!\!\!\perp Y\|Z)_p)$	$\neg I_p(\underline{X}\|\underline{Z}\|\underline{Y})$
$CI_p(\underline{X};\underline{Y}\|\underline{Z})$	$\underline{X}$ és $\underline{Y}$ beavatkozási függetlensége $\underline{Z}$ feltétellel p esetében
$\prec$	(részleges) sorrendezés
$\prec^c$	a változók egy teljes sorrendezése
$\prec^G$	adott G irányított körmentes gráffal kompatibilis sorrendek halmaza
$\prec(n)$	n objektum sorrendjeinek (permutációinak) a halmaza
$G, \underline{\theta}$	Bayes-háló struktúrája és paraméterei
$G^{\sim}$	G irányított körmentes gráf esszenciális gráfja
$\mathcal{G}(n)/\mathcal{G}^k(n)$	n csomópontú maximum k szülőjű DAG-ok halmaza
$\mathcal{G}^{\prec}$	adott $\prec$ sorrenddel kompatibilis DAG-ok halmaza
$\mathcal{G}^G$	adott G DAG-gal megfigyelési ekvivalens DAG-ok halmaza
$\sim$	kompatibilitási reláció
$\mathrm{pa}(X_i, G) \sim \prec$	$\mathrm{pa}(X_i, G)$ szülői halmaz kompatibilis $\prec$ sorrendezéssel
$\mathrm{MB}_p(X_i)$	Markov-takarója X_i-nek p-ben
$\mathrm{pa}, \mathrm{pa}(X_i, G)$	szülői változók halmaza, X_i szüleinek halmaza G-ben
pa_{ij}	a j. konfigurációja a szülői értékeknek egy sorrendben
$\mathrm{bd}(X_i, G)$	X_i szüleinek, gyerekeinek és gyerekei egyéb szüleinek halmaza G-ben
$\mathrm{MBG}(X_i, G)$	a Markov-takaró algráfja X_i-nek G-ben
$\mathrm{MBM}(X_i, X_j, G)$	a Markov-takaróbeliség relációja
n	valószínűségi változók száma
k	maximális szülőszám DAG-okban
N	mintaszám
V	összes valószínűségi változók száma
Y	válasz, kimeneteli, függő változó

[1] További konvenciók az egyes fejezetekben jelöltek.

$N_+/N_{\ldots,+,\ldots}$	$N_i/N_{\ldots,i,\ldots}$ megfelelő összegei
$D\|X$	X változóhalmazra szűkített adathalmaz
$\|\|$	kardinalitás
$1()$	indikátorfüggvény
f', f''	f függvény első és második deriváltjai
A^T	A mátrix transzponáltja
$\underline{x} \cdot \underline{y}$	$\underline{x}$ és $\underline{y}$ vektorok skalárszorzata
ξ^+/ξ^-	informatív/nem informatív információs kontextus
$\neg, \wedge, \vee, \neq, \rightarrow$	standard logikai operátorok
$\cap, \cup, \setminus, \Delta$	standard halmazműveletek
$KB \vdash_i \alpha$	α bizonyíthatósága KB-ből
Γ	a Gamma függvény
$\mathrm{Beta}(x\|\alpha, \beta)$	a Béta eloszlás sűrűségfüggvénye (pdf)
$\mathrm{Dir}(x\|\underline{\alpha})$	a Dirichlet eloszlás sűrűségfüggvénye
$\mathrm{N}(x\|\mu, \sigma)$	az egyváltozós normál eloszlás sűrűségfüggvénye
$\mathrm{N}(x\|\underline{\mu}, \underline{\underline{\Sigma}})$	a többváltozós normál eloszlás sűrűségfüggvénye
BD,BD_e	Bayesian Dirichlet prior, megfigyelési ekvivalens BD prior
BD_{CH}	Bayesian Dirichlet (BD) prior 1 hiperparaméterekkel
$\mathrm{BD}_{\mathrm{eu}}$	megfigyelési ekvivalens és uniform BD prior
$L(\underline{\theta}; D_N)$	$p(D_N\|\underline{\theta})$ likelihood függvénye
$H(X, Y)$	X és Y entrópiája
$I(X; Y)$	X és Y kölcsönös információja
$\mathrm{KL}(X\|\|Y)$	X és Y Kullback–Leibler divergenciája
$H(X\|\|Y)$	X és Y keresztentrópiája
$L_1(,)$, $L_2(,)$	az abszolútértékbeli (Manhattan) négyzetes (euklidészi) távolságok
$L_0(,)$	0-1 veszteség
$\mathcal{O}()/\Theta()$	aszimptotikus, nagyságrendi felső és alsó határ

Rövidítések

ROC	Receiver Operating Characteristic (ROC) görbe
AUC	ROC-görbe alatti terület
BMA	bayesi modell átlagolás
BN	Bayes-háló
DAG	irányított körmentes gráf
FSS	jegykiválasztási probléma
MAP	maximum a posteriori
MI	kölcsönös információ
ML	maximum likelihood
MBG	Markov-határ gráf
MB	Markov-takaró
MBM	Markov-takaróbeliség
(MC)MC	(Markov-láncos) Monte Carlo
NBN	naiv Bayes-háló

9. fejezet

Valószínűségi Bayes-hálók tanulása

A fejezetben bemutatjuk a valószínűségi gráfos modelleken (PGM) belül a Bayes-hálókhoz kapcsolódó paramétertanulást, majd a feltételes modellezés és teljes tárgyterületi modellezés kapcsolatát tisztázzuk, végül egy információelméleti megközelítést mutatunk be valószínűségi Bayes-hálók tanulására. A fejezet a Rejtett Markov Modellek paramétereinek tanulásának ismertetésével indul, amelyen keresztül bemutatjuk az Expectation-Maximization alapú paramétertanulást. Ismertetjük a feltételes modellek és PGM-k használatának pro és kontra érveit adatelemzési feladatokban, különös tekintettel a Naiv Bayes-hálók tanulásának szempontjából. A megkötések nélküli Bayes-hálók tanulására származtatunk egy pontszámot információelméleti, avagy „Minimum Description Length" formalizmus szerint. Az oksági Bayes-hálók tanulását, amely oksági vonatkozású priorokat is befogad, illetve oksági modelltulajdonságok tanulását is lehetővé teszi az Oksági Bayes-hálók tanulása fejezetben ismertetjük.

9.1. Paramétertanulás Rejtett Markov Modellekben

A valószínűségi gráfos modellek egyik népszerű alosztálya a Rejtett Markov Modellek (RMM), amelyek az eredeti beszédfelismerési és követési felhasználási területek mellett a biológiai szekvenciák elemzésében is egyre intenzívebben felhasználtak. A Bayes-háló formalizmust követve definíciójuk és a tárgyalásukban használt jelölésük a következő [39].

1. π jelöli a rejtett állapot szekvenciát, π_i jelöli az i. állapotot,
2. a_{kl} jelöli az állapotátmenet valószínűségeket $p(\pi_i = l|\pi_{i-1} = k)$ (egy extra 0 állapotot fenntartva az induló és végállapot számára),
3. $e_k(b)$ a kibocsájtási/megfigyelési valószínűségek $p(x_i = b|\pi_i = k)$.

A továbbiakban feltesszük, hogy az RMM-ekbeli változók diszkrét és véges változók. Az RMM-kben való következtetés típusait, benne az „előrefele" és „visszafele" RMM következtetési módszereket, a Valószínűségi döntéstámogatási rendszerek egy fejezete foglalja

össze, levezetésükért lásd [96]. A továbbiakban feltesszük, hogy ezek hatékony megoldása ismert, és az is, hogy általuk kiszámíthatóak a következőek:

1. „dekódolás": $\pi^* = \arg\max_\pi p(x, \pi)$,
2. szekvencia valószínűsége: $p(x) = \sum_\pi p(x, \pi)$ (avagy $p(x|M)$ „modell likelihood" vagy szűrés),
3. simítás/poszterior dekódolás: $p(\pi_i = k|x)$.

Feltételezve, hogy θ jelöli az RMM modell paramétereit, n független $x^{(1)}, \ldots, x^{(n)}$ megfigyelési szekvencia valószínűsége ekkor a következőképpen írható

$$p(x^{(1)}, \ldots, x^{(n)}|\theta) = \prod_{i=1}^{n} p(x^{(i)}|\theta). \tag{9.1}$$

Rögzített struktúrát, azaz rögzített megfigyelés- és állapotteret feltételezve két feladatot vizsgálunk meg. Bevezetési jelleggel elsőként megnézzük, hogy hogyan határozhatók meg a paraméterek a maximum likelihood elv szerint, ha ismertek a megfigyelésekhez tartozó állapotok, illetve, ha ezek nem ismertek. Ez utóbbi az úgynevezett Baum-Welch eljárás, amely az expectation-maximization eljárásra példa.

9.1.1. Paramétertanulás RMM-ekben ismert állapotszekvenciák esetében

Ismert állapotszekvenciák esetében az A_{kl} állapotátmenet számlálók és az $E_k(b)$ kibocsájtási számlálók segítségével a megfelelő a_{kl}, $e_k(b)$ relatív gyakoriságok közvetlenül adódnak

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}, \tag{9.2}$$

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}. \tag{9.3}$$

Emlékeztetőül idézzük, hogy a relatív gyakoriság maximum likelihood becslő a multinomiális mintavételhez. Tételezzük fel, hogy $i = 1, \ldots K$ kimenetelek egy multinomiális mintavételből származnak, amelynek paraméterei $\theta = \{\theta_i\}$, és jelölje $n = \{n_i\}$ a megfigyelt előfordulási számokat ($N = \sum_i n_i$). Ekkor

$$\log \frac{p(n|\theta^{ML})}{p(n|\theta)} = \log \frac{\prod_i (\theta_i^{ML})^{n_i}}{\prod_i (\theta_i)^{n_i}} = \sum_i n_i \log \frac{\theta_i^{ML}}{\theta_i} = N \sum_i \theta_i^{ML} \log \frac{\theta_i^{ML}}{\theta_i} > 0, \tag{9.4}$$

mivel $0 < KL(\theta^{ML}||\theta)$

$$-KL(p||q) = \sum_i p_i \log(q_i/p_i) \leq \sum_i p_i((q_i/p_i) - 1) = 0 \tag{9.5}$$

felhasználva, hogy $\log(x) \leq x - 1$.

Az a priori tudás kombinálására ad hoc módszerként jelent meg *pszeudoszámlálók* vagy *a priori számlálók* fogalma, ami kis mintás esetben a becslések robusztusságának növelésére is felhasznált

$$A'_{kl} = A_{kl} + r_{kl}, \tag{9.6}$$
$$E'_k(b) = E_k(b) + r_k(b), \tag{9.7}$$

amelynek elméleti hátterét az Oksági Bayes-hálók tanulása fejezetben tárgyaljuk.

9.1.2. E-M alapú paramétertanulás RMM-ekben ismeretlen állapotszekvenciák esetében

Egy közelítő módszer ismeretlen állapotszekvenciák esetében, hogy akár a priori vagy akár közelítő módszerekkel, mint a Viterbi algoritmus, rekonstruáljuk a legrelevánsabbnak vélt vagy legvalószínűbb állapotszekvenciákat, amelyet követően már a közvetlen becslési eljárás alkalmazható. Ez az ötlet iteratívan is használható, amely a következő módszerre vezet: becsüljük meg a várható értékét az A_t, E_t számlálóknak adott θ_t mellett, majd újrabecsülve frissítsük θ_{t+1} paramétereket ezen A_t,E_t... várható értékek alapján.

Elsőként is vegyük észre, hogy a $k \to l$ átmenetek valószínűsége az x szekvencia i. pozíciójában közvetlenül számítható a hatékonyan számolható „előrefele" és „visszafele" RMM következtetési módszerekből:

$$p(\pi_i = k, \pi_{i+1} = l|x) \tag{9.8}$$
$$= \frac{p(\overbrace{x_1, \ldots, x_i, \pi_i = k}^{f_k(i)}, x_{i+1}, \overbrace{\pi_{i+1} = l, x_{i+2}, \ldots, x_L}^{b_l(i+1)})}{p(x)} = \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1)}{p(x)}. \tag{9.9}$$

Ekkor a várható értéke egy adott átmenet valószínűségnek és kibocsátási valószínűségnek rendre a következő:

$$A_{kl} = \sum_j \frac{1}{p(x^{(j)})} \sum_i f_k^{(j)}(i) a_{kl} e_l(x_{i+1}^{(j)}) b_l^{(j)}(i+1) \tag{9.10}$$

$$E_k(b) = \sum_j \frac{1}{p(x^{(j)})} \sum_{i|x_i^{(j)}=b} f_k^{(j)}(i) b_k^{(j)}(i). \tag{9.11}$$

A módszer konvergenciáját az biztosítja, hogy az iteráltan elvégzett következtetés és paraméterbecslés valójában az úgynevezett E-M módszercsaládba tartozik. Általános bemutatásához megmutatjuk a fő ötletét, amely adatelemzési/optimalizálási feladatokban is gyakran használt, az eredeti, bizonyítottan jó működést adó feltételein túl. Az E-M módszercsalád fő célja az itteni jelölést használva, hogy a megfigyelt x és hiányzó π esetében határozza meg a maximum likelihood paramétereket

$$\theta^* = \arg\max_\theta \log(p(x|\theta)). \tag{9.12}$$

A megközelítés felfogható egy olyan módszernek, amikor a megoldás analitikusan és hatékonyan számolhatóan adódik, ha a hiányzó adat létezik, és ezen módszer a hiányzó adatok rögzített adatokkal való „pótlása" és hiányzó adatok kiátlagolása között helyezkedik el. Működésének központi eleme a „várható adat log-likelihood"

$$Q(\theta|\theta_t) = \sum_{\pi} p(\pi|x,\theta_t)\log(p(x,\pi|\theta)) \tag{9.13}$$

iterált javítása. Kihasználva, hogy

$$p(x,\pi|\theta) = p(\pi|x,\theta)p(x|\theta), \tag{9.14}$$

írhatjuk, hogy

$$\log(p(x|\theta)) = \log(p(x,\pi|\theta)) - \log(p(\pi|x,\theta)). \tag{9.15}$$

Ekkor $p(\pi|x,\theta_t)$-val szorozva és π felett szummázva

$$\log(p(x|\theta)) = \underbrace{\sum_{\pi} p(\pi|x,\theta_t)\log(p(x,\pi|\theta))}_{Q(\theta|\theta_t)} - \sum_{\pi} p(\pi|x,\theta_t)\log(p(\pi|x,\theta)). \tag{9.16}$$

Mivel a likelihoodot szeretnénk növelni, ez ekvivalens ennek a különbségnek a növelésével.

$$\log(p(x|\theta)) - \log(p(x|\theta_t)) = Q(\theta|\theta_t) - Q(\theta_t|\theta_t) + \underbrace{\sum_{\pi} p(\pi|x,\theta_t)\log(\frac{p(\pi|x,\theta_t)}{p(\pi|x,\theta)})}_{KL(p(\pi|x,\theta_t)||p(\pi|x,\theta))} \tag{9.17}$$

Kihasználva, hogy $0 \geq KL(p||q)$, adódik a következő

$$\log(p(x|\theta)) - \log(p(x|\theta_t)) \geq Q(\theta|\theta_t) - Q(\theta_t|\theta_t). \tag{9.18}$$

Az **általánosított E-M** eljárás csupán egy jobb θ megválasztásán alapszik $Q(\theta|\theta_t)$ tekintetében, amely folytonos paramétertér esetében garantált, hogy aszimptotikusan egy lokális vagy globális maximumhoz konvergál (sajnos diszkrét esetekben viselkedésére nincs garancia, bár gyakran használt, például lásd [48]). A standard **E-M** eljárás javító lépésében maximalizálás szerepel

$$\theta_{t+1} = \arg\max_{\theta} Q(\theta|\theta_t). \tag{9.19}$$

Visszatérve az RMM paramétertanulásra, az E-M módszer alkalmazása a következő. Egy adott π állapot- és x megfigyelés szekvencia valószínűsége a következő:

$$p(x,\pi|\theta) = \prod_{k=1}^{M}\prod_{b}[e_k(b)]^{E_k(b,\pi)}\prod_{k=0}^{M}\prod_{l=1}^{M} a_{kl}^{A_{kl}(\pi)}. \tag{9.20}$$

Ezt felhasználva $Q(\theta|\theta_t) = \sum_\pi p(\pi|x,\theta_t)\log(p(x,\pi|\theta))$ átírható a következőképpen:

$$Q(\theta|\theta_t) = \sum_\pi p(\pi|x,\theta_t)\sum_{k=1}^{M}\sum_b E_k(b,\pi)\log(e_k(b)) + \sum_{k=0}^{M}\sum_{l=1}^{M} A_{kl}(\pi)\log(a_{kl}). \tag{9.21}$$

Mivel A_{kl} és $E_k(b)$ várható értéke π-k felett egy adott x esetében

$$E_k(b) = \sum_\pi p(\pi|x,\theta_t)E_k(b,\pi) \;\; \mathrm{A_{kl}} = \sum_\pi \mathrm{p}(\pi|\mathrm{x},\theta_\mathrm{t})\mathrm{A_{kl}}(\pi), \tag{9.22}$$

így elsőként π-k felett elvégezve az összegzést kapjuk, hogy

$$Q(\theta|\theta_t) = \sum_{k=1}^{M}\sum_b E_k(b)\log(e_k(b)) + \sum_{k=0}^{M}\sum_{l=1}^{M} A_{kl}\log(a_{kl}). \tag{9.23}$$

Ekkor kihasználva, hogy A_{kl} és $E_k(b)$ hatékonyan számolható az előre és hátra algoritmusokkal az aktuális θ_t esetében, és a_kl és $b_k(l)$ alkotják az új paramétereket θ , $Q(\theta|\theta_t)$-t maximalizálja a következő:

$$a^0_{kl} = \tfrac{A_{ij}}{\sum_k A_{ik}}, \tag{9.24}$$

mivel az A-kat tartalmazó tagnál a különbség így írható

$$\sum_{k=0}^{M}\sum_{l=1}^{M} A_{kl}\log(\tfrac{a^0_{kl}}{a_{kl}}) = \sum_{k=0}^{M}(\sum_{l'} A_{kl'})\sum_{l=1}^{M} a^0_{kl}\log(\tfrac{a^0_{kl}}{a_{kl}}), \tag{9.25}$$

amelyik éppen a K-L távolság, tehát nem negatív.

9.2. Naiv Bayes-hálók tanulása

A Bayes-hálók modellosztály másik nagyon népszerű családja a Naiv Bayes-hálók (NBN) (lásd Valószínűségi gráfos modellek PGM fejezete). Az NBN-ek feltevéseinek széles körű elfogadhatósága mellett a tanulásban is központi szerepet töltenek be robusztusságuk miatt, mivel mintegy hidat képeznek az úgynevezett feltételes modellek felé. Elsőként ezt a határvonalat tisztázzuk, megvizsgálva az NBN-ek paraméter- és struktúratanulását is, a bayesi kontextusban is.

9.2.1. A bayesi feltételes modellezés

A *feltételes megközelítés* célja a bizonytalan reláció modellezése az Y kimeneti és az X bemeneti változók között. Egyéb elnevezési konvenciók a válasz, kimenetel, függő és prediktor, magyarázó vagy független változók. A bizonytalan reláció modellezésére hasonló

axiomatikus megközelítéssel adódik a valószínűségi megközelítés, amelyben fő cél ismét predikciós

$$p(Y_{N+1}|X_{N+1},(X_1,Y_1),\ldots,(X_N,Y_N)). \tag{9.26}$$

Analóg módon, egy feltételes felcserélhetőségi feltevésből származtatható egy parametrikus bayesi megközelítés, hasonlóan támogatva a „mintha" értelmezését a bayesi megközelítésnek [34]

$$p(y_1,\ldots,y_N|x_1,\ldots,x_N)=\int(\prod_{i=1}^{N}p(y_i|\theta(x_i)))p(\theta(x))\,\mathrm{d}\theta(x), \tag{9.27}$$

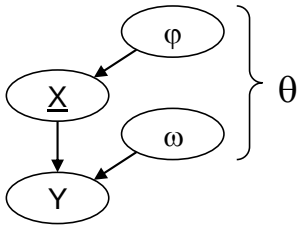
ahol $p(\theta(x))$ egy prior a $p(y_i|\theta(x_i))$-t specifikáló parametrikus család felett.

Alapvető eltérés a tárgyterületi vagy rendszeralapú modellezéshez képest, hogy a bemeneti változók közti összefüggések nincsenek modellezve. Ezen adatok hiányos volta az egyik legnagyobb kihívás a feltételes modellezés gyakorlati alkalmazásánál, amire közelítő, a teljes tárgyterületi modellezést elkerülő, de legalábbis minimalizáló megoldások széles skálája jött létre (lásd Hiányos adatok kezelése fejezet). Egy elméleti nehézség a feltételes modellek bayesi alkalmazása terén az úgynevezett konjugált prior hiánya és az ezzel összekapcsolódó probléma, hogy a tárgyterületi modellezéstől eltérően a paraméterek szintje itt analitikusan nem kezelhető (lásd Oksági Bayes-hálók tanulása fejezet).

A tárgyterületet leíró függetlenségi modell meghatározza a validitását az esetleges feltételes modellezésnek, ami statisztikai mintaszám komplexitás és számítási komplexitás tekintetében mindenképpen előnyösebb. Ez a bayesi megközelítésben is megnyilvánul, és általános esetben a $p(G),p(\theta|G)$ struktúra és paraméter priorok például Bayes-hálókra nem dekomponálódik priorokra a $p(Y|X)$ feltételes modell szerint. Például még ha tárgyterületi, a priori kényszerek miatt is a kimeneti változó nem lehet szülő, akkor is függhet ennek a változónak a modellje a többi változó közti függésektől

$$p(y|x)=\sum_{G}p(G)\int_{\Theta}p(y|x,\theta,G)\,\mathrm{d}p(\theta). \tag{9.28}$$

Egy pragmatista bayesi megközelítésben, a 9.27 egyenletben azt tesszük fel, hogy az Y X-től való függésére vonatkozó elvárások függetlenek más tárgyterületi függésektől. Ennek formalizálására tételezzük fel, hogy a teljes (Y,X)-ra vonatkozó megfigyeléseket egy prior $p(\theta)$ és egy mintavételi $p(Y,X|\theta)$ eloszlás definiálja, ahol a θ paraméter (ϕ,ω)-ra bontható, ahol a ϕ paraméter X-hez tartozik (azaz $X\not\perp\!\!\!\perp\phi$ és $(X\perp\!\!\!\perp\omega|\phi)$), és ω pedig $Y|X$-hez tartozik (azaz $Y\not\perp\!\!\!\perp\{X,\omega\}$ and $(Y\perp\!\!\!\perp\phi|\omega,X)$). A feltételes megközelítés ekkor formálisan azt teszi fel, hogy $\omega\perp\!\!\!\perp\phi$, azaz a priorok ilyen dekomponálását, amit a 9.1 ábrán lévő Bayes-háló is illusztrál.



9.1. ábra. A bayesi feltételes modellezés feltevéseinek Bayes-hálós modellje. A bemeneti és kimeneti változókat X és Y jelöli, a hozzájuk tartozó paramétereket ϕ és ω (amelyek együtt alkotják θ-t).

Az ezen feltevés szerint dekomponált priorok jelentik a feltételes megközelítés alapját, mivel teljes megfigyelés esetén a posteriorok is dekomponálódnak

$$\begin{aligned} p(\theta|x,y) \triangleq p(\omega,\phi|x,y) &\propto p(x,y|\omega,\phi)p(\omega,\phi) && (9.29)\\ &= p(y|x,\omega)p(x|\phi)p(\omega|\phi)p(\phi) && (9.30)\\ &= p(y|x,\omega)p(\omega)p(x|\phi)p(\phi) && (9.31)\\ &\propto p(\omega|x,y)p(\phi|x). && (9.32)\end{aligned}$$

Eszerint, ha csak a feltételes modellre szeretnénk következtetni, azaz ω poszteriorra, akkor a feltételes megközelítésben

$$\begin{aligned} p(\omega|x,y) &\propto \int_\phi p(y|x,\omega,\phi)p(x|\phi)p(\omega|\phi)p(\phi)\,\mathrm{d}\phi && (9.33)\\ &= p(y|x,\omega)p(\omega). && (9.34)\end{aligned}$$

9.2.2. Bayes-hálók tanulása feltételes modellként

A hiányos adatkezelés vagy a priori elvárások miatt is a teljes tárgyterületi modell tanulása sok esetben egy minta- és számításigényes, de legalább univerzális választásnak tűnhet. Azonban viszonylagosan kis mintáknál, akár a bayesi keretben is a később tárgyalt jegytanulási módszerekben, a teljes modell tanulása szisztematikusan eltér a feltételes modell tanulásától. Ennek megértéséhez fontoljuk meg a teljes adatra vonatkozó likelihood-t egy olyan G,θ modellnél, amely az $\underline{X},Y$-t tartalmazza [49]:

$$\begin{aligned} LL(G,\theta;D_N) &= \log p(D_N|G,\theta) && (9.35)\\ &= CLL_Y(G,\theta;D_N) + \sum_{i=1}^{N}\log p(\underline{X}_i|G,\theta) && (9.36)\end{aligned}$$

ahol

$$CLL_Y(G,\theta;D_N) \;=\; \sum_{i=1}^{N}\log p(Y_i|G,\theta,\underline{X}_i). \tag{9.37}$$

Az első tag a feltételes adat log-likelihood $CLL_Y(G,\theta;D_N)$, amelyik kizárólagosan meghatározza az osztályozást (diszkrét esetet feltételezve). A másik tag annak a következménye, hogy nem feltételes modellosztállyal dolgozunk, és mint látni fogjuk, mind becslési bias-t és nagyobb statisztikai érzékenységet is okoz. Elsőként is, a Bayes-háló általános volta miatt az osztályozás szempontjából optimális $\arg\max_\theta CLL_Y(G,\theta;D_N)$ paraméterek nem egyenlőek a tárgyterületi modell maximum likelihood paramétereivel $\arg\max_\theta LL_Y(G,\theta;D_N)$, sem a paraméterek várható értékeivel $\mathrm{E}_{p(\theta|G,D_N)}[\theta]$ (valamely $p(\theta)$ priorral) (ez a nem faktorizálódó normalizációs konstansok miatt is fellép). Ez alól csak az jelent kivételt, ha a kimeneti Y változó levél csomópont [49]. Másodsorban, a $CLL_Y(G,\theta;D_N)$ tag dominált az $n-1$ analóg taggal a bemeneti változóknál [49].

9.2.3. Naiv Bayes-hálók teljesítménye osztályozásban és regresszióban

Az NBN-eknek rengeteg sikeres alkalmazása van, akár numerikus vagy folytonos bemeneti változókkal, mind az osztályozási és mind a regressziós keretben. Mivel az NBN-ek függetlenségi feltevése szerint az $Y(=X_0)$ központi (gyökér/szülő) változó és az n darab $X_1,\ldots,X_n$ bemeneti változóra fennáll, hogy $X_i \perp\!\!\!\perp \{X_j : j\neq i\}|Y$, így adódik a jól ismert eredmény

$$\log P(y|\underline{X}) = \sum_{i=1}^{n}\log p(X_i|y) + \log P(y) - \log P(\underline{X}), \tag{9.38}$$

amely azt mutatja, hogy diszkrét esetben az NBN egy lineáris osztályozó (diszkriminátor). Azonban, ha folytonos változók is megengedettek, akkor nem lineáris és egymástól elválasztott régiók is megjelenhetnek döntési tartományoknak [38, 36]. Az NBN-ek sikeres alkalmazása osztályozásban, amikor az osztályozás a regressziósbecslésen definiált egy küszöbértékkel, elméleti kutatásokban is vizsgált [45, 36]. J. Friedman mutatta meg, hogy a D_N adat statisztikai hatása az osztályozási hibán strukturálisan különbözik a regressziós becslés L_2 predikciós hibájától. A jól ismert „bias-variance" megközelítés szerint az L_2 hiba a frekventista megközelítésben regressziós feladatban egy bias és egy variancia tagra bontható [55],

$$\mathrm{E}_{D_N}[y-\hat{f}_{D_N}(\underline{x})]^2 \;=\; \mathrm{E}_{D_N}[f(\underline{x})-\hat{f}_{D_N}(\underline{x})]^2 + \mathrm{E}[\underbrace{y-f(\underline{x})}_{\epsilon\text{ noise}}]^2 \tag{9.39}$$

$$\mathrm{E}_{D_N}[f(\underline{x})-\hat{f}_{D_N}(\underline{x})]^2 \;=\; \underbrace{(f(\underline{x})-\mathrm{E}_{D_N}[\hat{f}_{D_N}(\underline{x})])^2}_{\text{bias}} + \underbrace{\mathrm{E}_{D_N}[\mathrm{E}_{D_N}[\hat{f}_{D_N}(\underline{x})]-\hat{f}_{D_N}(\underline{x})]^2}_{\text{variance}}.$$

Bináris osztályozásnál a bias tényező lépcsős jellegű küszöbhöz kötött $\text{sign}(\frac{1}{2} - f)(E\hat{f} - \frac{1}{2})$ és a varianciával skálázott $\text{var}(\hat{f})$. Ez azt eredményezi, hogy a regresszióbeli varianciája (azaz komplexitása) a függvényosztálynak nagyobb hangsúlyt kap és a bias-e (azaz kapacitása) pedig kisebbet. Máshogyan fogalmazva, az olyan egyszerű modellosztály, mint a naiv BN, alkalmatlan lehet regresszióra a szigorú korlátai miatt (bias), de kiváló osztályozásra, mivel varianciája kicsi [45].

9.2.4. Naiv Bayes-hálók kiterjesztései

Az NBN modellek szigorú feltevését több módon is próbálták enyhíteni, így növelve az alkalmazhatóságát: a *Tree Augmented NBN* (*TAN*) és a kontextuális multi-háló [49], az *Augmented NBN* osztályozó [36, 70, 81], stb. A TAN modellosztály egy skálázható kiterjesztést jelent, így azt ismertetjük. Ebben a modellosztályban (az Y kimeneti és n darab $X_1, \ldots, X_n$ bemeneti változó felett) egy teljes NBN definiált, amely modellstruktúra pluszban egy teljes fát is tartalmaz a bemeneti változók csomópontjai között. Ez lehetővé tesz globális és lokális szenpontok optimalizálásával $n-1$ él beillesztését a maximális szülőszámot $k \leq$ 2-n tartva [49]. Kihasználva, hogy teljes D_N adat esetén egy adott G struktúránál a $\underline{\theta}^*$ maximum likelihood paramétereket használva a log-likelihood felírható így (levezetésért lásd Oksági modellek tanulása fejezet)

$$\log p(D_N|G, \underline{\theta}^*) = N \sum_{i=0}^{n} I(X_i; \text{Pa}(X_i, G)) - N \sum_{i=1}^{n} H(X_i), \tag{9.40}$$

felhasználva még a kölcsönös információra vonatkozó láncszabályt $I(X; Y, Z) = I(X; Z) + I(X; Y|Z)$, az első, struktúrától függő összeg átírható a következőképpen (külön gyűjtve a szülőre vonatkozó konstans tagokat)

$$\sum_i I_{\hat{p}_{D_N}}(X_i; Y) + \sum_{i:\text{pa}(X_i)\geq 1} I_{\hat{p}_{D_N}}(X_i; X_{\text{pa}(X_i)}|Y). \tag{9.41}$$

Ez azt mutatja, hogy a maximum likelihood TAN-t úgy találhatjuk meg, hogy a második összegzést maximáljuk, bevéve a maximális feltételes kölcsönös információjú éleket a „korrelációs" fába a bemeneti változók között. Mivel az élekre a feltételes kölcsönös információ $\mathcal{O}(n^2 N)$ időben kiszámolható, amiből ezt az értéket, mint élsúly kezelve *maximális feszítőfa* (MWST) konstruálható $\mathcal{O}(n^2 \log n)$ időben, ez egy nagyon hatékony algoritmust jelent [49]. A módszer hátránya, hogy egy teljes fát épít fel akkor is, ha a bemeneti változók az NBN feltevés szerint feltételesen függetlenek, de ez tesztekkel el is kerülhető, illetve komplexitásnövelő hatása mintakomplexitás emelkedés tekintetében mérsékelt. Általánosabb módszerek nemcsak fát, hanem általános DAG struktúrát is megengednek a bemeneti változók közös szülő/gyökér változóval feltételes függéseinek reprezentálására, például k maximális szülőszámot előírva, azonban $1 < k$-ra már NP-teljes az optimális bemeneti változók feletti DAG megkeresése, így általános tanulási eljárások szükségesek.

9.2.5. Teljes modellátlagolás NBN-ek felett

Az NBN-ekkel való következtetés hatékonyságát, változók számában lineáris voltát egy pontparametrizáció mellett vizsgáltuk meg. Ez a lineáris komplexitás a paraméterek bayesi kezelése esetében is gyakran megmarad, mivel ha $p(\Theta|G)$ jelöl egy adott struktúra melletti paramétereloszlást, akkor egy értékek szintjén történő következtetésben a paramétertér felett átlagolás

$$\mathrm{E}_{\underline{\Theta}}[p(\underline{y}|\underline{x},\underline{\Theta})] = \bar{p}(\underline{y}|\underline{x}) \tag{9.42}$$

adott, praktikus feltételek mellett analitikusan elvégezhető (lásd Oksági modellek tanulása fejezet). Egy meglepő eredmény szerint ez a lineáris komplexitás még akkor is megtartható, ha a struktúrák felett is átlagolni kell, azaz ha a 2^n struktúrák feletti $p(G)$ eloszlás követi a tárgyterületi függetlenségek szerinti dekompozíciót, akkor létezik egy olyan „szuper-(pont)paraméterezés", amely ezt is képes reprezentálni [30]. Elsőként is idézzük fel, hogy $\underline{x}$ teljes megfigyelés esetében a feltételes eloszlás $p(y|\underline{x})$ kiszámolható a $p(Y,\underline{x})$ együttes eloszlásból $\mathcal{O}(n)$ időben, az NBN-eket definiáló szorzatalakból

$$p(Y|\underline{x},\underline{\theta},G) \propto p(Y,\underline{x},\underline{\theta},G) = p(Y)\prod_{i=1}^{n} p(x_i|\,\mathrm{pa}(X_i,G)) = \prod_{i=0}^{n} p(x_i|\,\mathrm{pa}(X_i,G)). \tag{9.43}$$

Másodsorban, Dirichlet paraméterpriorokat használó paraméterfüggetlenség esetén a paraméterek feletti integrálás elkerülhető a lokális modellekbeli valószínűségek várható értékeinek használatával (lásd 10.8 egyenlet).

$$p(y,\underline{x}) = \sum_G p(G)\int p(y,\underline{x},\underline{\theta},G)\,\mathrm{d}p(\underline{\theta}) = \sum_G p(G)\prod_{i=0}^{n} p(x_i|\,\mathrm{pa}(X_i,G)). \tag{9.44}$$

Teljes D_N adatok esetén ez a tulajdonság a poszteriorokra is teljesül (lásd 10.13 egyenlet).

Harmadrészt, strukturális modularitást feltéve, ha a struktúrák felett prior a $p(Y \to X_i \in Edges(G))$ élek a priori élvalószínűségeinek szorzataként definiálható, amely a DAG kényszerrel való kompatibilitása miatt egy normalizált priort definiál. Elméletileg a következtetésbe belépő, akár részleges $\underline{x}$ is befolyásolja a poszteriort, de ezt figyelmen kívül hagyjuk.

Ekkor a struktúrák feletti poszteriorra az alábbi szorzatalak adódik (lásd 10.31 egyenlet, a normalizáció minden változóra függetlenül végezhető el, megfontolva a két lehetséges esetet, hogy Y szülő-e vagy nem)

$$p(G|D_N) = \prod_{i=0}^{n} p(\mathrm{Pa}(X_i,G)|D_N), \tag{9.45}$$

és jelölje p_i^1 $p(\mathrm{Pa}(X_i,G) = Y|D_N)$-t és p_i^0 $p(\mathrm{Pa}(X_i,G) \neq Y|D_N)$-t (kihasználva, hogy $p_0^0 = p_Y^0 = 0$).

Kombinálva a 9.43,a 9.44 és a 9.45 egyenleteket kapjuk, hogy

$$
\begin{aligned}
& p(y, \underline{x}|D_N) && (9.46)\\
&= \sum_G \prod_{i=0}^{n} p(pa(X_i)^G, D_N) p(x_i|pa(X_i)^G, D_N) && (9.47)\\
&= \sum_{\substack{pa^G_{X_1}=Y \\ pa^G_{X_1}\neq Y}} \cdots \sum_{\substack{pa^G_{X_n}=Y \\ pa^G_{X_n}\neq Y}} \prod_{i=0}^{n} p(pa(X_i)^G|D_N) p(x_i|pa(X_i)^G, D_N) && (9.48)\\
&= \prod_{i=0}^{n} p_i^0 p(x_i|pa(X_i)^G \neq Y, D_N) + p_i^1 p(x_i|pa(X_i)^G = Y, D_N) && (9.49)\\
&= \prod_{i=0}^{n} p^*(x_i|pa(X_i)^G = Y, D_N), && (9.50)
\end{aligned}
$$

ahol az utolsó lépésben felhasználtuk az összeg-szorzat felcserélését. Ez azt mutatja, hogy egy pontparaméterezés konstruálható egy teljes NBN modellre $\mathcal{O}(n)$ időben, amellyel a 9.44 egyenletbeli prediktív következtetés — 2^n modellstruktúra és paramétertér feletti átlagolás — $\mathcal{O}(n)$ időben elvégezhető.

9.3. Egy információelméleti pontszám Bayes-háló tanulásához

Az általános valószínűségi Bayes-hálók tanulásához a jelen fejezetben egy információelméleti pontszámot mutatunk be. A Bayes-hálók általános, gazdagabb, akár oksági vonatkozású priorokat is befogadó tanulását az Oksági Bayes-hálók tanulása fejezetben ismertetjük.

A frekventista paradigmában tételezzünk fel teljes, diszkrét, azonos eloszlású, független (i.i.d.) mintákat és definiáljunk egy maximum likelihood pontszámot a következőképpen

$$ML(G; D_N) = max_{\underline{\theta}} p(D_N|G, \underline{\theta}). \tag{9.51}$$

Adott G struktúránál a 9.4 egyenlet szerint ezt a $\theta^*_{ijk} = N_{ijk}/N_{ij+}$ relatív gyakoriságok maximálják, ahol N_{ijk} jelöli az x_k érték és q_j szülői konfiguráció együttes előfordulásának számát X_i változó és $Pa(X_i)$ szülői halmazára nézve (N_{ij+} az értelemszerű összeget jelenti). Behelyettesítve ezt a maximum likelihood paramétert kapjuk, hogy

$$
\begin{aligned}
ML(G; D_N) &= p(D_N|G, \underline{\theta}^*) = \prod_{l=1}^{N} \prod_{i=1}^{n} p(x_i^{(l)}|pa_i^{(l)}) && (9.52)\\
&= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(\frac{N_{ijk}}{N_{ij+}}\right)^{N_{ijk}}. && (9.53)
\end{aligned}
$$

Logaritmust véve, összegzési sorrendet cserélve és N-nel bővítve adódik, hogy

$$\log(ML(G;D_N)) = N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\frac{N_{ij+}}{N}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N_{ij+}}\log(N_{ijk}/N_{ij+}). \tag{9.54}$$

Felhasználva a feltételes entrópia definícióját $H(Y|X) = \sum_x p(x)\sum_y p(y|x)\log(p(y|x))$, az entrópiára vonatkozó láncszabályt $H(X,Y) = H(Y|X) + H(X)$ és a kölcsönös információ definícióját $I(Y;X) = H(Y) - H(Y|X)$ [26], ez átírható a következőképpen:

$$\begin{aligned}
\log(ML(G;D_N)) &= -N\sum_{i=1}^{n}H(X_i|Pa(X_i,G)) && (9.55)\\
&= -NH(X_1,\ldots,X_n) && (9.56)\\
&= N\sum_{i=1}^{n}I(X_i;Pa(X_i,G)) - N\sum_{i=1}^{n}H(X_i). && (9.57)\\
& && (9.58)
\end{aligned}$$

Ez azt mutatja,hogy a definiált maximum likelihood pontszámot maximalizáló Bayes-háló értelmezhető úgy, mint ami entrópiát minimalizál, avagy az általa elérhető kódolás is a legrövidebb hosszúságú, ami a „Minimum Description Length" tömörítési elvet tükrözi (lásd 9.56), másrészt úgy is, hogy maximalizálja a szülők és a gyerekeik közti kölcsönös információt (lásd 9.57, ahol a struktúrától független tagokat értelemszerűen nem számítanak). Nagyon érdekes kapcsolat, hogy a változók azon sorrendje, amely ilyen értelemben maximális meghatározottságot mutat, az mennyiben használható fel oksági sorrendek kikövetkeztetésére [37].

A definiált pontszám struktúratanulásban való felhasználásához azonban figyelembe kell venni a kölcsönös információ monotonitását, nevezetesen, hogy ha $Pa(X_i) \subset Pa(X_i)'$, akkor $I(X_i;Pa(X_i) \leq I(X_i;Pa'(X_i)$ [26], azaz a teljes DAG maximalizálja a pontszámot. Ezt egy komplexitás regularizációs tag tudja kezelni, amely a „Bayesian information criterion" (BIC) pontszám esetében a következő teljes pontszámot eredményezi (származtatását és egyéb pontszámokat az Oksági Bayes-hálók fejezetben tárgyalunk [77, 12, 22, 50, 62])

$$BIC(G;D_N) = \log(ML(G;D_N)) - \frac{1}{2}dim(G)\log(N), \tag{9.59}$$

ahol $dim(G)$ a szabad paraméterek számát jelöli.

10. fejezet

Oksági Bayes-hálók tanulása

A fejezetben bemutatjuk a bayes statisztikai keretben történő tanulását Bayes-hálós modelleknek, ami lehetővé teszi oksági modellek részletes kiértékelését akár on-line tanulási keretben, oksági vonatkozású priorok felhasználását, oksági modelltulajdonságok bayesi tanulását. Az oksági priorok mellett beavatkozási adatok felhasználása is biztosítja oksági aspektusok koherens felhasználását, illetve a Bayes-hálós modellosztály oksági értelmezéssel felruházható modelltulajdonságai. Elsőként a „predictive sequential" (prequential) kiértékelési módszertant foglaljuk össze. Majd rögzített struktúra mellett bemutatjuk a paraméterszint analitikus kezelésének lehetőségét, ami paraméterpriorok és adatok koherens, analitikus kombinálását teszik lehetővé. Áttekintjük az oksági priorokat, majd származtatjuk a struktúrákra vonatkozó megfigyeléseken alapuló poszteriort és annak beavatkozási adatra vonatkozó adaptálását, amely így oksági modellek tanulását teszi lehetővé. Összefoglaljuk a pontszámok kisszámú mintánál jelentkező különbségeit, aszimptotikus tulajdonságait, ideértve végtelen adat (orákulum alapú) tanulási módszereket.

10.1. Bayesi következtetés és tanulás rögzített oksági struktúra esetén

A bayesi megközelítésben egy G rögzített oksági struktúra esetén a $p(\underline{\theta}|G)$ paraméter eloszlás tekinthető adottnak. A bayes tanulás — pragmatista szempontból elsődlegesnek — tekintett alapvető célja a tárgyterületi megfigyelésekkel való következtetés, amelynek egy eredménye a $p(\underline{y}|\underline{x},\underline{\Theta})$ valószínűségi változó. Amint látni fogjuk, ha a $p(\underline{\theta}|G)$ paramétereloszlást megfelelően választjuk meg, akkor az eredmény egyszerű analitikus formában adódik, például igaz lesz, hogy $p(y|\underline{x},\underline{\Theta})$ egy úgynevezett Dirichlet eloszlást követ $Np_0(y,\underline{x})$ hiperparaméterekkel. Ennek tisztázásához és jelentőségének a megértéséhez ismerkedjünk meg a konjugált eloszlások fogalmával, amely mind a priorok hatékony megadásához, a prediktív következtetéshez és a paraméterekre vonatkozó következtetéshez is alapvető fontosságú.

10.1.1. Definíció. *A priori* eloszlások egy $\mathcal{F}$ családja, amely $p(\theta)$ paraméterekkel adott konjugált a $p(x|\theta)$ mintavételi eloszlás családra, ha a $p(\theta|x)$ *a posteriori* eloszlások is $\mathcal{F}$-

beliek.

Az analitikus azonosságon túl további előny, hogy az exponenciális eloszlásokhoz tartozó konjugált priorok esetében a megadásuk úgynevezett *hiperparaméter*ekkel történik és a poszterior pedig ezen hiperparaméterek aktualizált értékei szerint, ahol az aktualizálás tipikusan az adat rövid leíró statisztikáiból származtatható (lásd [57, 10]). Ebben az esetben a hiperparamétereknek gyakran egy intuitív értelmezése is lehetséges, így a priorhoz és a poszteriorhoz tartozó hiperparaméterek a korábban és az együttesen látott megfigyelések egyszerű statisztikai jellemzői. Erre példa a következő, ahol a θ paraméter eloszlásának az α, β hiperparamétereinek egy egyszerű számláló értelmezése van.

10.1.1. Példa. Jelölje x az összegét n darab független, azonos eloszlású (i.i.d.) Bernoulli mintavételnek (0 és 1 értékek felett), azaz egy binomiális eloszlást követ. Ha a paraméterpriort egy Beta eloszlás definiálja, akkor a poszterior is Beta eloszlás lesz aktualizált hiperparaméterekkel:

$$p(x|\theta) = Bin(x|n,\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \tag{10.1}$$

$$p(\theta) = Beta(\theta|\alpha,\beta) = c\theta^{\alpha-1}(1-\theta)^{\beta-1} \text{ where } c = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{10.2}$$

$$p(\theta|x) = \frac{p(\theta)p(x|\theta)}{p(x)} = c'\theta^{\alpha-1+x}(1-\theta)^{\beta-1+n-x} = Beta(\theta|\alpha+x,\beta+n-x).$$

A következő példa pedig a bayesi prediktív következtetést mutatja be r diszkrét érték esetén (amelyek legyenek az $1,\ldots,r$ értékek).

10.1.2. Példa. Jelölje $D_n = \{X_i; i = 1, 2, \ldots, n\}$ a megfigyelések sorozatát, amely i.i.d. multinomiális mintákat tartalmaz r diszkrét értékből. Legyen a prior $p(\theta)$ egy Dirichlet prior $\underline{\alpha} = (\alpha_1,\ldots,\alpha_r)$ hiperparaméterekkel, és legyen $\alpha_+ = \sum_k \alpha_k$:

$$\mathrm{Dir}(\theta|\underline{\alpha}) = c\prod_k \theta_k^{\alpha_k-1}, \text{ where } c = \frac{\Gamma(\alpha_+)}{\prod_k \Gamma(\alpha_k)}. \tag{10.3}$$

Ez a prior konjugált a multinomiális mintavételre, a poszterior prediktív eloszlás egy aktualizált Dirichlet $\underline{\alpha}_i$ hiperparaméterekkel az i. lépésben, és a poszterior prediktív eloszlás az x_i értékre (azaz a $\mathrm{E}[\theta_{x_i}]$ marginális poszterior)

$$
\begin{aligned}
p(x_i|x_1,\ldots,x_{i-1}) &= \int p(x_i|\underline{\theta})\mathrm{Dir}(\underline{\theta}|\underline{\alpha}_i)\,\mathrm{d}\underline{\theta} \\
&= c\int \theta_{x_i}\prod_k \theta_k^{\alpha_{i,k}-1}\,\mathrm{d}\underline{\theta},\ \text{ahol } c=\frac{\Gamma(\alpha_{i,+})}{\prod_k\Gamma(\alpha_{i,k})} \\
&= c\int\prod_k \theta^{\alpha_{i+1,k}-1}\mathrm{d}\theta, \text{ahol } \alpha_{i+1,k}=\alpha_{i,k}, \text{except } \alpha_{i+1,x_i}=\alpha_{i,x_i}+1 \\
&= \frac{\Gamma(\alpha_{i,+})}{\Gamma(\alpha_{i+1,+})}\frac{\prod_k\Gamma(\alpha_{i+1,k})}{\prod_k\Gamma(\alpha_{i,k})} \\
&= \frac{\alpha_{i,x_i}}{\alpha_{i,+}}, \qquad (10.4)
\end{aligned}
$$

így a D_n marginális valószínűsége egy $\underline{\theta}\sim\mathrm{Dir}(\underline{\alpha}_1)$ prior és az $k=1,\ldots,r$ értékek n_k előfordulásaival

$$
\begin{aligned}
p(x_1,\ldots,x_n) &= \prod_{i=1}^{n} p(x_i|x_1,\ldots,x_{i-1}) && (10.5)\\
&= \frac{\prod_{k=1}^{r}(\alpha_{1,k}\ldots(\alpha_{1,k}+n_k))}{\alpha_{1,+}\ldots(\alpha_{1,+}+n)} && (10.6)\\
&= \frac{\Gamma(\alpha_{1,+})}{\Gamma(\alpha_{1,+}+n)}\frac{\prod_{k=1}^{r}\Gamma(\alpha_{1,k}+n_k)}{\prod_{k=1}^{r}\Gamma(\alpha_{1,k})}. && (10.7)
\end{aligned}
$$

Egy oksági modell esetében a paraméterek feletti eloszlás — amint azt láttuk az Oksági modellek fejezetben — követheti maga a struktúra, azaz az általa kifejezett függetlenségi viszonyokat is. Ezen esetben adódik a következő központi eredmény. Ha a $p(\underline{\theta}|G)$ eloszlás Dirichlet eloszlások által speficikált és eleget tesz a paraméterfüggetlenségi feltételnek, tetszőleges hiperparaméterekkel(!), akkor a $\bar{p}(X_1,\ldots,X_n)$ marginális eloszlás a tárgyterületi értékek szerint a következőképpen egyszerűsíthető:

$$
\begin{aligned}
\bar{p}(x_1,\ldots,x_n) &= \int p(x_1,\ldots,x_n,\underline{\theta}_1,\ldots,\underline{\theta}_n)\prod_{i=1}^{n}p(\underline{\theta}_i)\,\mathrm{d}\underline{\theta} && (10.8)\\
&= \prod_{i=1}^{n}\int p(x_i|\,\mathrm{pa}(x_i),\underline{\theta}_i)p(\underline{\theta}_i)\,\mathrm{d}\underline{\theta}_i && (10.9)\\
&= \prod_{i=1}^{n}\bar{p}(x_i|\,\mathrm{pa}(x_i)), && (10.10)
\end{aligned}
$$

ahol $\bar{p}(x_i|\,\mathrm{pa}(x_i))$ a lokális valószínűségek várható értékei [103, 102, 27]. Azaz, az egyes szülői konfigurációknál szereplő paraméterek várható értékei zárt formában adódnak a 10.4 egyenlet szerint

$$
\bar{p}(X_i=k|\,\mathrm{pa}(X_i)=\mathrm{pa}_{ij})=\mathrm{E}_{\underline{\Theta}_i}[p(X_i=k|pa_{ij},\underline{\Theta}_i)]=\mathrm{E}_{\underline{\Theta}_{ij}}[\Theta_{ijk}]=N_{ijk}/N_{ij}.
$$

A $\bar{p}(X_1, \ldots, X_n)$ zárt megoldás létének jelentőségét az adja, hogy a paraméter eloszlások ezen eseteiben egy rögzített oksági struktúránál bármely bayesi következtetés ekvivalens módon elvégezhető a várható értékek szerinti pontparametrizációval, a paramétertér feletti bayesi átlagolás helyett [103, 25]

$$\mathrm{E}_{\underline{\Theta}}[p(\underline{y}|\underline{x}, \underline{\Theta})] = \bar{p}(\underline{y}|\underline{x}). \tag{10.11}$$

Az általános esetben ilyen „szuper"-paraméterezés nem lehetséges, azaz amikor a struktúrák felett is adott egy $p(G)$ eloszlás és kapcsolódó $p(\underline{\theta}|G)$ paraméter eloszlások

$$\bar{p}(\underline{y}|\underline{x}) = \mathrm{E}_{p(G)}[\mathrm{E}_{p(\underline{\theta}|G)}[p(\underline{y}|\underline{x}, \underline{\theta}, G)]]. \tag{10.12}$$

Ekkor például Monte Carlo módszerekkel következtethetünk a $p(\underline{y}|\underline{x}, \underline{\Theta}, G)$ véletlen változó várható értékére és bizonyossági intervallumaira.

A „szuper"-paraméterezés azonban fennmarad, sőt nagyon könnyen aktualizálható is újabb teljes megfigyelések esetében. Tételezzünk fel egy x teljes megfigyelést, paraméter függetlenséget és Dirichlet priorokat $\underline{\theta}_{ij} \sim \mathrm{Dir}(\alpha_{ij1}, \ldots, \alpha_{ijr_i})$ $i = 1, \ldots, n$ és $j = 1, \ldots, q_i$ esetére (ahol r_i jelöli az X_i változó értékeit, q_i pedig a számát a $\mathrm{pa}(X_i, G)_j = \mathrm{pa}_{ij}$ szülői konfigurációknak az X_i változónál a G oksági strukturában). Ekkor a $\underline{\theta}_{ij_0}$ paraméterek eloszlása a „megfigyelt" esetekben, amelynek j_0 az indexe $pa_i(x)$-k között így adódik

$$\begin{aligned} p(\underline{\theta}|x) &= \frac{\prod_{i=1}^{n} p(x_i|pa_i(x), \underline{\theta}_{ij_0}) p(\theta_{ij_0})}{p(x)} \prod_{i=1}^{n} \prod_{j \neq j_0} p(\theta_{ij}) && (10.13) \\ &\propto \prod_{i=1}^{n} \theta_{ij_0 x_i} \mathrm{Dir}(\theta_{ij_0}|\underline{\alpha}_{ij_0}) && (10.14) \\ &\propto \prod_{i=1}^{n} \mathrm{Dir}(\theta_{ij_0}|\alpha_{ij_0 1}, \ldots, \alpha_{ij_0 x_i} + 1, \ldots, \alpha_{ij_0 r_i}), && (10.15) \end{aligned}$$

amely azt mutatja, hogy a paraméter poszterior megőrzi a paraméter függetlenségi tulajdonságot, és a bayesi aktualizálást a teljes esetben előforduló, „megfigyelt" Dirichlet eloszlások hiperparamétereinek aktualizálása jelenti ($j \neq j_0$ esetében, azaz a többi $\underline{\theta}_{i_0 j}$ paramétereknek az eloszlása nem változik).

10.2. A prekvenciális modellkiértékelés

A tudásmérnöki modellkonstrukció és a modellek teljes strukturális tanulása között helyezkedik el a modellek kiértékelése és adaptálása, amely a szakértő tudás és adatok kombinációján és finom egyensúlyán alapul. Ennek segítésére született meg a „predictive sequential" (prequential) keretrendszer, amely az adatok szekvenciális természetét, akár a tárgyterület nem stacioner, azaz változó voltát is képes kezelni [31, 32].

10.2.1. Általános és valószínűségi előrejelző rendszerek vizsgálata

A prekvenciális keretben az előrejelző rendszer és annak részeinek kvantitatív minősítése az előrejelzések miatt veszteségek alapján történik. Feltételezett, hogy a rendszer a $D = X_1, X_2, \ldots$ szekvencia megfigyelése közben minden egyes lépésben egy F_{n+1} előrejelzést ad az addig megfigyelt $D_n = \{x_1, \ldots, x_n\}$ szekvencia alapján. Az előrejelzést az $S(F_{n+1}, x_{n+1})$ pontszám minősíti és a teljes S minősítési pontszám ezek összegeként definiált. Esetünkben az előrejelző rendszer valószínűségi, amikor a modellosztály megadása, egy prior megadása és egy mintavételi eloszlás megadása iteratívan definiálja az előrejelzést a $p_n(X_n|X_1, \ldots, X_{n-1})$ feltételes eloszlásokon keresztül, amelyek a lépésenkénti poszterior prediktív eloszlások. Itt is feltesszük, hogy a bizonytalan mennyiségeknek r diszkrét $s_1, \ldots, s_r$ értéke lehetséges és hogy a $\underline{q_n}$ előrejelzései a valószínűségi rendszernek a $\underline{q_n} = p_n(X_n|x_1, \ldots, x_{n-1})$ poszterior prediktív eloszlásokon alapulnak.

Ha a poszterior előrejelzést („report”-olását) a döntéselméleti keretben mint akciót, választást értelmezzük, akkor a pontszám egy veszteségfüggvényként értelmezhető $S(\underline{q}, s_k)$ és $\underline{q_n}$ a minimális veszteségű előrejelzésnek kell lennie

$$\arg\min_{\underline{q}} \sum_{k=1}^{r} S(\underline{q}, s_k) p_n(X_n = s_k|x_1, \ldots, x_{n-1}). \tag{10.16}$$

Megmutatható, hogy becsületesség („igaz elvárások közlése”), folytonosság („arányos büntetés a hibákért”) és dekomponálhatóság (a büntetés a konkrét {előrejelzés-megfigyelés} páron alapul) feltevések maguk után vonják a logaritmikus pontszámfüggvényt, $S(\underline{q}, s_k) = A \log(q_k) + B_k$, ahol $A < 0$ és B_k tetszőleges konstansok [10].

A logaritmikus veszteségfüggvény esetében a valószínűségi előrejelző rendszereknél az adódik, hogy

$$S_n(p_n(X_n|x_1, \ldots, x_{n-1}), x_n) = -\log(p_n(X_n = x_n|x_1, \ldots, x_{n-1})), \tag{10.17}$$

aminek több hasznos következménye is van.

Elsőként is D_n adatnál az összpontszám marginális adata likelihood logaritmusa, ami független a sorrendtől is, így használható az adatok nem szekvenciális, „batch” felhasználásánál is:

$$\begin{aligned} S &= \sum_{i=1}^{n} S_i(p_i(X_i|x_1, \ldots, x_{i-1}), x_i) \\ &= -\log \prod_{i=1}^{n} p_i(x_i|x_1, \ldots, x_{i-1}) \\ &= -\log p(x_1, \ldots, x_n). \end{aligned} \tag{10.18}$$

Másodsorban, az összehasonlító modellkiértékelés segítésére, az M előrejelző rendszert az M^{ref} rendszerhez hasonlítva adódik, hogy

$$\exp(S - S^{ref}) = \frac{p(x_1, \ldots, x_n|M)}{p(x_1, \ldots, x_n|M^{\text{ref}})} \tag{10.19}$$

ami a Bayes faktor.

Végezetül a logaritmikus veszteségfüggvény esetében egy $\underline{q} \neq \underline{p}$ előrejelzésnek a várható vesztesége a keresztentrópia $H(p\|q) = \mathrm{KL}(p\|q) + H(p)$.

10.2.2. Bayes-hálók prekvenciális vizsgálata

A prekvenciális vizsgálat a bayesi filozófia szerint a konkrét adaton alapul és akár mintánkénti kiértékelést is lehetővé tesz, vagy akár változó eloszlásból származó mintákon történő kiértékelést is biztosít [102, 27]. A következőkben feltesszük, hogy a valószínűségi előrejelző rendszer egy oksági Bayes-hálóként definiált, ahol a struktúra rögzített és paraméterek eloszlása Dirichlet priorokkal adott a paraméterfüggetlenség feltételezésével. Ekkor a következő modellpontszámokat vezethetjük be.

A *globális (modell) monitor* („követő") az $M = (G, \underline{\theta})$ Bayes-háló model átfogó teljesítményét jelzi a D_N adathalmazon:

$$\begin{aligned} S(M; D_N) &= \sum_{l=1}^{N} -\log p(x^{(l)}|x^{(1)}, \ldots, x^{(l-1)}, M) \\ &= -\log p(x^{(1)}, \ldots, x^{(N)}|M). \end{aligned} \tag{10.20}$$

Mint látni fogjuk a 10.27 és a 10.30 egyenletekben, ez a modell likelihood, ami így írható

$$S(M; D_N) = -\log \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+} + n_{ij+})} \frac{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + n_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})}. \tag{10.21}$$

A struktúra szerinti dekomponálás szellemében — a 10.31 egyenlet szerint — különböző egyéb monitorok is hasznosak a modell részletesebb vizsgálatára.

A (feltétel nélküli) *csomópont monitor* az M modell kontextusában az X_i változót követi:

$$S(X_i; D_N) = -\log \prod_{l=1}^{N} p(x_i^{(l)}|x^{(1)}, \ldots, x^{(l-1)}, M). \tag{10.22}$$

Két variánsa a csomópont monitornak a feltételes csomópont monitorok, amelyekben a feltétel vagy minden más változót vagy csak a szülő változókat tartalmazza. Az előbbit nevezték „feltételes csomópont monitornak" [102]),

$$S(\mathrm{MBG}(X_i, G); D_N) = -\log \prod_{l=1}^{N} p(x_i^{(l)}|x^{(l)} \setminus \{X_i\}, x^{(1)}, \ldots, x^{(l-1)}). \tag{10.23}$$

Az oksági modellezésben fontos szülői halmaz követésére szolgál a *mechanizmus monitor*:

$$S(\mathrm{Pa}(X_i, G); D_N) = -\log \prod_{l=1}^{N} p(x_i^{(l)}|\,\mathrm{pa}(X_i) = \mathrm{pa}_i^{(l)}, x^{(1)}, \ldots, x^{(l-1)}). \tag{10.24}$$

A mechanizmuson belüli vizsgálatra alkalmas a (szülői) *konfiguráció monitor*, amely egy konkrét pa_{ij} viszony paraméterezését követi:

$$S(\mathrm{pa}_{ij}; D_N) = -\log \prod_{l=1}^{N} p(x_i^{(l)}|pa_{ij}, x^{(1)}, \ldots, x^{(l-1)})^{1(\mathrm{pa}_i^{(l)}=\mathrm{pa}_{ij})}. \quad (10.25)$$

10.3. Oksági struktúrák tanulása

Az oksági modellek paramétertanulása és modellkiértékelése után áttekintjük az oksági struktúrák tanulását is. Elsőként a kényszer alapú módszereket illusztráljuk röviden, majd a Valószínűségi Bayes-hálók tanulása fejezetben levezetett információelméleti struktúrapontszám mellé az oksági priorok befogadását is lehetővé tevő bayesi poszterior származtatását mutatjuk be. Ismertetjük az optimalizálás eddig megismert elméleti korlátait. Végül oksági jegyek bayesi tanulására mutatunk példákat.

10.3.1. Kényszer alapú struktúratanulás

A kényszer alapú struktúratanulási algoritmusok lehetőség szerint minimális számú függetlenségi tesztet végrehajtva próbálnak olyan Bayes-háló struktúrát találni, amely az adatokban megjelenő függetlenségi viszonyokat hűen reprezentálja [90, 58, 104] (minimális függetlenségi térkép, lásd Valószínűségi gráfos modellek fejezet). Ezekre az algoritmusokra példa az „Inductive Causation" (IC) algoritmus, amely egy stabil eloszlást tételez fel és ekkor helyes megoldást ad:

1. *Váz:* Konstruáljuk meg az irányítatlan gráfot (vázat) úgy, hogy $X, Y \in \underline{V}$ akkor legyen összekötve, ha $\forall S(X \perp\!\!\!\perp Y|S)_P$, ahol $S \subseteq \underline{V} \setminus \{X, Y\}$.

2. *v-struktúrák:* Irányítsuk $X \to Z \leftarrow Y$, ha X, Y nem szomszédosak, Z egy közös szomszéd és $\neg\exists S$ úgy, hogy $(X \perp\!\!\!\perp Y|S)_P$, ahol $S \subseteq \underline{V} \setminus \{X, Y\}$ és $Z \in S$.

3. *propagation:* Irányítsuk a maradék irányítatlan éleket úgy, hogy nem hozunk létre új v-struktúrát, sem irányított kört.

10.3.1. Tétel. *A következő szabályok szükségesek és elégségesek.*

R_1 *ha* $(a \not\to c) \wedge (a \to b) \wedge (b - c)$, *akkor* $b \to c$

R_2 *ha* $(a \to c \to b) \wedge (a - b)$, *akkor* $a \to b$

R_3 *ha* $(a - b) \wedge (a - c \to b) \wedge (a - d \to b) \wedge (c \not\to d)$, *akkor* $a \to b$

R_4 *ha* $(a - b) \wedge (a - c \to d) \wedge (c \to d \to b) \wedge (c \not\to b) \wedge (a - d)$, *akkor* $a \to b$.

Bár stabil eloszlás esetében a módszerek aszimptotikus adatmennyiségnél azonosan viselkednek, véges adatmennyiségnél nincs gyakorlati tanács a szignifikancia szintek kezelésére, sem a globálisan kiadódó modell átfogó szignifikancia szintjére. Mint látni fogjuk, ez egy NP-teljes feladat (lásd 10.3.5. tétel), azonban alacsony számítási igénye miatt és rejtett változókat is kezelő kiterjesztései miatt ez a megközelítés lokális oksági részstruktúrák kikövetkeztetésére egy vonzó lehetőség.

10.3.2. Pontszámok oksági struktúrák tanulására

A kényszer alapú megközelítéssel szemben a pontszám alapú struktúratanulás egy $S(G, D_N) : \{G \times D_N\} \to \mathcal{R}$ adat alapú függvényen és ennek a maximalizálásán, tehát optimalizálási eljárásokon alapul. A Valószínűségi Bayes-hálók tanulása fejezetben levezetett információelméleti struktúrapontszám mellé egy bayesi poszterior alapú pontszámot is származtatunk. Egy Bayes-háló struktúra poszteriorja a struktúra priornak és a modell likelihoodnak a szorzata

$$p(G|D_N) \propto p(G) \int p(D_N|\underline{\theta}, G) p(\underline{\theta}|G) \, \mathrm{d}\underline{\theta} = p(G) p(D_N|G). \tag{10.26}$$

A likelihood tényező zárt alakban történő származtatásához a korábbi, prekvenciális tárgyalásban is használt feltevéseket használjuk: N teljes megfigyelés, i.i.d. multinomiális mintavétel és hogy a Bayes-háló modell paraméter függetlenség feltevése mellett Dirichlet paraméter priorokkal specifikált [25, 102, 63]. Ezen feltevések mellett a megőrzött paraméter függetlenségi tulajdonság miatt lehetségesek az alábbi lépések:

$$\begin{aligned} p(x^{(1)}, \ldots, x^{(N)}|G) &= \prod_{l=1}^{N} \prod_{i=1}^{n} p(x_i^{(l)}|pa_i^{(l)}) && (10.27) \\ &= \prod_{i=1}^{n} \prod_{l=1}^{N} p(x_i^{(l)}|pa_i^{(l)}) && (10.28) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{l=1}^{N} p(x_i^{(l)}|pa_{ij})^{1(\mathrm{pa}_{ij}=\mathrm{pa}_i^{(l)})}, && (10.29) \end{aligned}$$

ahol $pa_i^{(l)}$ jelöli az X_i szülői konfigurációjának értékét az l. mintában. A marginális valószínűsége az adatnak egyetlen Dirichlet prior és multinomiális mintavétel esetén a 10.4, a 10.4 és a 10.5 egyenletekben lett származtatva. Ha r_i jelöli az X_i változó értékeinek számát, α_{ijk} a kezdeti Dirichlet hiperparamétereket, és n_{ijk} az X_i változó, pa_{ij} szülői konfigurációjának és r_k értékének az előfordulási számát, akkor egy adott X_i változó j. szülői konfigurációjára függetlenül adódik, hogy

$$\prod_{l=1}^{N} p(x_i^{(l)}|pa_{ij},G)^{1(\mathrm{pa}_{ij}=\mathrm{pa}_i^{(l)})} = \frac{\prod_{k=1}^{r_i}(\alpha_{ijk}\dots(\alpha_{ijk}+n_k))}{\alpha_{ij+}\dots(\alpha_{ij+}+n)} \quad (10.30)$$

$$= \frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+}+n_{ij+})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+n_{ijk})}{\Gamma(\alpha_{ijk})}.$$

Együttesen így az adódik, hogy ha a prior teljesíti a strukturális modularitás feltételét (azaz például szülői halmazonként definiált), akkor egy Bayes-háló struktúra poszteriorja a következő szorzat formájában adódik

$$p(G|D_N) \propto \prod_{i=1}^{n} p(\mathrm{Pa}(X_i,G))S(X_i,\mathrm{Pa}(X_i,G),D_N), \text{ ahol} \quad (10.31)$$

$$S(X_i,\mathrm{Pa}(X_i,G),D_N) = \prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+}+n_{ij+})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+n_{ijk})}{\Gamma(\alpha_{ijk})}.$$

Ezt *Bayesian Dirichlet* pontszámnak nevezik, és ha az $\underline{\alpha}$ kezdeti hiperparaméterek kielégítik azokat a feltételeket, amelyek egy megfigyelési ekvivalencia osztályon belüli megkülönböztethetetlenséget követelnek meg, akkor BD_e jelöli [63]. Ha a kezdeti $\underline{\alpha}$ hiperparaméterek konstans 1 értékűek, akkor $\mathrm{BD}_{\mathrm{CH}}$ jelöli [25]. Ha a kezdeti hiperparaméterek a lokális multinomiális modell paraméterei összámának reciproka, akkor jele $\mathrm{BD}_{\mathrm{eu}}$ [15, 63]. A kapcsolódó pontszámfüggény a következő $\mathrm{BD}(G;D_N) = \log(p(G,D_N))$. Beavatkozásos adatoknál az Oksági modellek fejezetben bevezetett „do" szemantika szerint annyit változik ez a pontszám, hogy a beállított változókhoz tartozó szorzatok nem jelennek meg [58].

A pontszámok fontos tulajdonsága a megfigyelési ekvivalencia osztályon belüli neutralitásuk, aszimptotikus helyességük és alacsony mintaszám esetében mutatott elfogultságuk (bias-ük).

10.3.1. Definíció. Egy pontszámfüggvény $S(G;D_N)$ *pontszám ekvivalens*, ha egy megfigyelési ekvivalencia osztályba tartozó struktúrákhoz azonos értéket rendel bármely adat esetén [63].

10.3.2. Tétel ([63])**.** *A $BD_e(G;D_N)$ pontszámfüggvény likelihood ekvivalens, azaz ha G_1,G_2 megfigyelési ekvivalens, akkor $p(D_N|G_1) = p(D_N|G_2)$. Továbbá, ha a struktúra prior akauzális (azaz egyenlő bármely ilyen G_1,G_2-re), akkor a BD_e pontszámfüggvény pontszám ekvivalens [63].*

Ennek megfelelően a BD pontszám használható mind oksági, mind valószínűségi Bayes-hálók tanulására. Az oksági megközelítésben, ha az adat megfigyelési, akkor az ekvivalencia osztályon belüli(!) megkülönböztetésre csak a priori hordoz oksági információt, amelynek hatása a mintaszám növekedésével aszimptotikusan eltűnik. A nem likelihood ekvivalens BD pontszámok ekvivalencia osztályon belüli különbségtétele hasonlóan eltűnik növekvő mintaszámnál.

A BIC pontszám ekvivalenciája a közvetlen következménye annak, hogy az egy ekvivalencia osztályba eső DAG-ok szabad paramétereinek száma megegyezik [12, 22, 21].

10.3.3. Tétel ([21]). *A $BIC(G; D_N)$ pontszámfüggvény pontszám ekvivalens.*

Az aszimptotikus optimalitást a következő tétel mondja ki [12].

10.3.4. Tétel ([12]). *Tételezzük fel, hogy a $p(\underline{V})$ eloszlás szigorúan pozitív és stabil, amelynek G_0 egy perfekt térképe. Legyen a $p(G)$ Bayes-háló struktúrák feletti eloszlás szigorúan pozitív. Ezek mellett, ha D_N egy i.i.d. adat $p(\underline{V})$ szerint, akkor bármely G $\underline{V}$ feletti struktúrára, amely nem perfekt térképe $p(\underline{V})$-nek, adódik, hogy*

$$\lim_{N\to\infty} \mathrm{BD}_e(G_0; D_N) - \mathrm{BD}_e(G; D_N) = -\infty \text{ illetve, hogy} \tag{10.32}$$

$$\lim_{N\to\infty} \mathrm{BIC}_e(G_0; D_N) - \mathrm{BIC}_e(G; D_N) = -\infty. \tag{10.33}$$

PAC tanulási eredményekért lásd [12, 50], illetve a paraméterek mintakomplexitásáért [28].

Sajnos a pontszámok aszimptotikus helyessége ellenére véges mintán a maximális pontszámú struktúrák tipikusan különböznek [12]).

10.3.3. Az optimalizálás nehézsége struktúratanulásban

A bevezetett struktúra pontszámok hatékonyan számolhatóak, n változós és N teljes minta esetében a feltevések mellett $\mathcal{O}(nN)$ időben. A DAG (vagy PDAG) térben történő optimalizálás nehézségét a következő két tétel jelzi (feltéve, hogy $P \neq NP$). Az első az NP-nehéz voltát jelzi egy olyan Bayes-háló megkeresésének, amely a megfigyelt függetlenségeket hűen reprezentálja [12].

10.3.5. Tétel ([12]). *Legyen $\underline{V}$ változóhalmaz feletti eloszlás $p(\underline{V})$. Tételezzünk fel egy orákulumot, amely $\mathcal{O}(1)$ időben megadja, hogy egy adott függetlenségi állítás teljesül-e p. Legyen $0 < k \leq |\underline{V}|$ és $s = \frac{1}{2}n(n-1) - \frac{1}{2}k(k-1)$. Ekkor annak eldöntése, hogy létezik-e olyan (akár nem minimális) Bayes-háló, ami p-t reprezentálja és éleinek száma s-nél kevesebb, az orákulum eléréseit tekintve NP-teljes.*

Egy másik tétel egy a pontszám szerint optimális Bayes-háló struktúra megkeresésének NP-nehéz voltát bizonyítja [22].

10.3.6. Tétel ([22]). *Jelölje $\underline{V}$ a változóhalmazt, efeletti D_N a teljes adathalmazt és $S(G, D_N)$ a pontszámfüggvényt. Ekkor NP-teljes annak eldöntése, hogy van-e olyan G_0 Bayes-háló struktúra $\underline{V}$ felett, hogy minden csomópont G_0-ban legfeljebb $1 < k$ szülője van és $c \leq S(G_0, D_N)$, ahol $c \in \mathcal{R}$.*

Speciális esetként adódik a $k = 1$ eset (azaz fák és polifák tanulása), amelynek polinom tanulási idejű megoldása van [89, 22] (alkalmazásáért lásd Valószínűségi Bayes-hálók tanulása fejezet). A probléma NP-nehéz volta marad az ekvivalencia osztályok terében is [22, 72].

A feladat nehézsége miatt optimalizálási módszerek sokaságát dolgozták ki, egyszerű lokális keresési eljárásoktól szimulált lehűtési sémákat használó módszerekig. Kiemelkedő

fontosságú és az oksági jegytanulás szempontjából is nagyon fontos felismerés volt a változók sorrendjének explicit reprezentálása, mivel egy adott sorrenddel vett feltétel mellett polinom időben megtalálható a sorrenddel topológiai sorrend szerint kompatibilis legnagyobb pontszámú struktúra (maximális szülőszámot korlátozva) [25].

Oksági jegyek bayesi tanulását a Bioinformatika Oksági adatelemzés és következtetés jegyzetben tárgyaljuk.

Irodalomjegyzék

[1] Y. Abramson and Y. Freund. Active learning for visual object recognition. Technical report, University of California San Diego, 2003.

[2] E. Acuna and C. Rodriguez. The treatment of missing values and its effect on classifier accuracy. In D. Banks, F. McMorris, R. Frederick, P. Arabie, and W. Gaul, editors, *Classification, Clustering, and Data Mining Applications*, Studies in Classification, Data Analysis, and Knowledge Organisation, pages 639–647. Springer Berlin Heidelberg, 2004.

[3] P.D. Allison. *Missing Data.* Sage Publications, Thousand Oaks, CA, 2001.

[4] Márta Altrichter, Gábor Horváth, Béla Pataki, György Strausz, Gábor Takács, and József Valyon. *Neurális hálózatok.* Panem könyvkiadó Kft, Budapest, 2006.

[5] FJ Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973.

[6] A. Antos. Valószínűségi becslés- és döntéselmélet. In *Valószínűségi döntéstámogató rendszerek*, Biotechnológia és bioinformatika, egyetemi jegyzet 1. fejezet. Typotex Kiadó, Budapest, Hungary, 2014. In print.

[7] A. Antos, V. Grover, and Cs. Szepesvári. Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411(29–30):2712–2728, June 17 2010. Special Issue for ALT 2008. Available online 10 April 2010. A previous version of this paper appeared at ALT 2008.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[9] John T. Behrens. Principles and procedures of exploratory data analysis. *Psychological Methods*, 2(2):131–160, 1997.

[10] J. M. Bernardo. *Bayesian Theory.* Wiley & Sons, Chichester, 1995.

[11] Nedret Billor, Ali S Hadi, and Paul F Velleman. Bacon: blocked adaptive computationally efficient outlier nominators. *Computational Statistics & Data Analysis*, 34(3):279–298, 2000.

[12] R. R. Bouckaert. *Bayesian Belief Networks: From construction to inference*. Ph.D. Thesis, Dept. of Comp. Sci., Utrecht University, Netherlands, 1995.

[13] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

[14] Jan Christian Bryne, Eivind Valen, Man-Hung Eric Tang, Troels Marstrand, Ole Winther, Isabelle da Piedade, Anders Krogh, Boris Lenhard, and Albin Sandelin. Jaspar, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucleic Acids Research*, 36(suppl 1):D102–D106, 2008.

[15] W. L. Buntine. Theory refinement of Bayesian networks. In *Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence (UAI-1991)*, pages 52–60. Morgan Kaufmann, 1991.

[16] A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In J. Kivinen, Cs. Szepesvári, E. Ukkonen, and T. Zeugmann, editors, *22nd International Conference on Algorithmic Learning Theory, ALT 2011, Proceedings*, volume 6925 of *LNCS/LNAI*, pages 189–203, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] A. Carpentier, R. Munos, and A. Antos. Minimax-optimal strategy for stratified sampling for Monte Carlo. *Journal of Machine Learning Research*, 2014. Accepted conditioned on minor revisions.

[18] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[19] P. Chaudhuri and P. Mykland. On efficient designing of nonlinear experiments. *Statistica Sinica*, 5:421–440, 1995.

[20] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[21] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proc. of 11th Conference on Uncertainty in Artificial Intelligence (UAI-1995)*, pages 87–98. Morgan Kaufmann, 1995.

[22] D. M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: Search methods and experimental results. In *Proceedings of Fifth Conference on Artificial Intelligence and Statistics*, pages 112–128, 1995.

[23] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[24] Dianne Cook and Deborah F Swayne. *Interactive and dynamic graphics for data analysis: with R and GGobi*. Springer, 2007.

[25] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[26] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley & Sons, 2001.

[27] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Springer-Verlag, New York, 1999.

[28] S. Dasgupta. The sample complexity of learning fixed-structure Bayesian networks. *Machine Learning*, 29:165–180, 1997.

[29] S. Dasgupta and J. Langford. Active learning. Tutorial T7 on Twenty-Sixth International Conference on Machine Learning, June 14, 2009.

[30] D. Dash and G. F. Cooper. Exact model averaging with naive Bayesian classifiers. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 91–98, 2002.

[31] A. P. Dawid. *Encylopedia of Statistical Sciences*, volume 1, chapter Prequential Analysis, pages 464–470. Wiley & Sons, 1997.

[32] A. P. Dawid and V. G. Vovk. Prequential probability: Principles and properties. *Bernoulli*, 5:125–162, 1999.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc., Series B*, 39(1):1–38, 1977.

[34] D. G. T. Denison, C. C. Holmes, B. K. Mallick, and A. F. M. Smith. *Bayesian Methods for Nonlinear Classification and Regression*. Wiley & Sons, 2002.

[35] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, NY, 1996.

[36] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[37] M. J. Druzdzel and H. Simon. Causality in Bayesian belief networks. In David Heckerman and Abe Mamdani, editors, *Proceedings of the 9th Conf. on Uncertainty in Artificial Intelligence (UAI-1993)*, pages 3–11. Morgan Kaufmann, 1993.

[38] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley & Sons, 2001.

[39] R. Durbin, S. R. Eddy, and A. Krogh anf G. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Chapman & Hall, London, 1995.

[40] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*, Chapman & Hall, 1993.

[41] P. Etore and B. Jourdain. Adaptive optimal allocation in stratified sampling methods. *Methodology and Computing in Applied Probability*, 2008. http://www.citebase.org/abstract?id=oai:arXiv.org:0711.4514.

[42] V. V. Fedorov. *Theory of Optimal Experiments.* Academic Press, 1972.

[43] G.S. Fishman. *Monte Carlo Concepts, Algorithms, and Applications.* Springer-Verlag, 1999.

[44] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering, and L. J. Jensen. STRING v9.1: Protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, 41(Database - issue):D808 – 15, 2013.

[45] J. H. Friedman. On bias, variance, 0/1-loss, and the curse of dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[46] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Fourteenth Inter. Conf. on Machine Learning (ICML97)*, pages 125–133. Morgan Kaufmann, SF, 1997.

[47] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proc. of the Fourteenth conf. on Uncertainty in AI (UAI).*, pages 129–138. Morgan Kaufmann, SF, 1998.

[48] N. Friedman. The Bayesian structural EM algorithm. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence(UAI-1998)*, pages 129–138. Morgan Kaufmann, 1998.

[49] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian networks classifiers. *Machine Learning*, 29:131–163, 1997.

[50] N. Friedman and Z. Yakhini. On the sample complexity of learning Bayesian networks. In *Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence (UAI-1996)*, pages 274–282. Morgan Kaufmann, 1996.

[51] Michael Friendly. A brief history of data visualization. In Chun-houh Chen, Wolfgang Hardle, and Antony Unwin, editors, *Handbook of Data Visualization*, chapter 2, pages 15–56. Springer Berlin Heidelberg, 2008.

[52] H. Gray Funkhouser. A note on a tenth century graph. *Osiris*, 1:pp. 260–262, 1936.

[53] Reinhard Furrer, Douglas Nychka, and Stephen Sain. fields: Tools for spatial data. *Rpackage version*, 6, 2009.

[54] Dani Gamerman and Hedibert Freitas Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition.* Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.

[55] S. Geman, S. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[56] A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge University Press, New York, 2007.

[57] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman & Hall, London, 1995.

[58] C. Glymour and G. F. Cooper. *Computation, Causation, and Discovery.* AAAI Press, 1999.

[59] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer, New York, NY, 2002.

[60] Jingrui He. *Analysis of rare categories.* Springer, 2012.

[61] Jingrui He and Jaime Carbonell. Coselection of features and instances for unsupervised rare category analysis. *Statistical Analysis and Data Mining*, 3(6):417–430, 2010.

[62] D. Heckermann. A tutorial on learning with Bayesian networks., 1995. Technical Report, MSR-TR-95-06.

[63] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[64] N.J. Horton and K.P Kleinman. Much ado about nothing. *The American Statistician*, 61(1):79–90, 2007.

[65] D.C. Howell. The treatment of missing data. In W. Outhwaite and S.P. Turner, editors, *The SAGE Handbook of Social Science Methodology*, pages 208–224. SAGE, London, 2007.

[66] Hsu, Sheng-Da and Lin, Feng-Mao and Wu, Wei-Yun and Liang, Chao and Huang, Wei-Chih and Chan, Wen-Ling and Tsai, Wen-Ting and Chen, Goun-Zhou and Lee, Chia-Jung and Chiu, Chih-Min and Chien, Chia-Hung and Wu, Ming-Chia and Huang, Chi-Ying and Tsou, Ann-Ping and Huang, Hsien-Da. miRTarBase: a database curates experimentally validated microRNA–target interactions, *Nucleic Acids Research* 2010. doi = 10.1093/nar/gkq1107

[67] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, May 1994.

[68] Alfred Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer Science+Business Media, LLC, 2009.

[69] Alan J. Izenman. *Modern Multivariate Statistical Techniques*. Springer Texts in Statistics. Springer New York, New York, NY, 2008.

[70] E. Keogh and M. Pazzani. Learning the structure of augmented Bayesian classifiers. *International Journal of Artificial Intelligence Tools*, 11(4):587–601, 2002.

[71] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, February 2000.

[72] T. Kocka and R. Castelo. Improved learning of Bayesian networks. In Jack S. Breese and Daphne Koller, editors, *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 269–276. Morgan Kaufmann, 2001.

[73] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *In Proceedings of the ICML*, pages 315–322, 2002.

[74] Ana Kozomara and Sam Griffiths-Jones. mirbase: integrating microrna annotation and deep-sequencing data. *Nucleic Acids Research*, 39(suppl 1):D152–D157, 2011.

[75] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[76] T. L. Lai and S. Yakowitz. Machine learning and nonparametric bandit theory. *IEEE Transactions on Automatic Control*, 40:1199–1209, 1995.

[77] W. Lam and F. Bacchus. Using causal information and local measures to learn Bayesian networks. In David Heckerman and Abe Mamdani, editors, *Proc. of the 9th Conference on Uncertainty in Artificial Intelligence (UAI-1993)*, pages 243–250. Morgan Kaufmann, 1993.

[78] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7–9):1578 – 1594, 2008. Progress in Modeling, Theory, and Application of Computational Intelligenc 15th European Symposium on Artificial Neural Networks 2007 15th European Symposium on Artificial Neural Networks 2007.

[79] T. Linder and G. Lugosi. *Bevezetés az Információelméletbe*. Tankönyvkiadó, Budapest, 1990. jegyzetszám: J5-1445.

[80] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley, New York, 1987.

[81] P. Lucas. Restricted Bayesian network structure learning. In H. Blockeel and M. Denecker, editors, *Proc. of 14th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'02)*, pages 211–218, 2002.

[82] Martin Maronna and MASS Suggests. Package 'robustbase'. 2009.

[83] Jean-Claude Masse, Jean-Francois Plante, and Maintainer Jean-Francois Plante. Package 'depth', 2009.

[84] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender. Transfac® and its module transcompel®: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34(suppl 1):D108–D110, 2006.

[85] R McGill, JW Tukey, and WA Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[86] Kari S McLeod. Our sense of Snow: the myth of John Snow in medical geography. *Social Science & Medicine*, 50(7-8):923–935, April 2000.

[87] J.W. Osborne. *Best Practices in Data Cleaning.* SAGE Publications, University of Louisville, KY, 2013.

[88] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326. IEEE, 2003.

[89] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Francisco, CA, 1988.

[90] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

[91] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013.

[92] C. Riggelsen. Learning bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. In *In Proceedings of the 6th SIAM Intl. Conf. on Data Mining.*, pages 130–140. SIAM, 2006.

[93] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer Texts in Statistics. Springer, 2004.

[94] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[95] Peter J Rousseeuw and Bert C Van Zomeren. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):633–639, 1990.

[96] S. Russel and P. Norvig. *Artificial Intelligence.* Prentice Hall, 2001.

[97] Ida Ruts and Peter J Rousseeuw. Computing depth contours of bivariate point clouds. *Computational Statistics & Data Analysis*, 23(1):153–168, 1996.

[98] Deepayan Sarkar. *Lattice: multivariate data visualization with R*. Springer, 2008.

[99] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[100] P. Sebastiani and M. Ramoni. Bayesian inference with missing data using bound and collapse. kmi-tr-58., 1997.

[101] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *Knowledge and Data Engineering, IEEE Transactions on*, 19(5):631–645, 2007.

[102] D. J. Spiegelhalter, A. Dawid, S. Lauritzen, and R. Cowell. Bayesian analysis in expert systems. *Statistical Science*, 8(3):219–283, 1993.

[103] D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed acyclic graphical structures. *Networks*, 20(.):579–605, 1990.

[104] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2001.

[105] W Stahel and M Maechler. robustx: experimental extraneous extraordinary. *Functionality for Robust Statistics. R package version*, pages 1–1, 2009.

[106] Peter D. Stenson, Edward V. Ball, Matthew Mort, Andrew D. Phillips, Katy Shaw, and David N. Cooper. *The Human Gene Mutation Database (HGMD) and Its Exploitation in the Fields of Personalized Genomics and Molecular Evolution*. John Wiley & Sons, Inc., 2002.

[107] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining*, pages 535–548. Springer, 2002.

[108] M.A. Tanner and W.H. Wong. The calculation of posterior distributions by data augmentation. *JASA*, 82(398):528–540, 1987.

[109] Martin Theus. Interactive data visualization using mondrian. *Journal of Statistical Software*, 7(11):1–9, 11 2002.

[110] Martin Theus and Simon Urbanek. *Interactive graphics for data analysis: principles and examples*. CRC Press, 2011.

[111] Luis Torgo and Maintainer Luis Torgo. Package 'dmwr'. 2013.

[112] L.-C. Tranchevent, F. B. Capdevila, D. Nitsch, B. De Moor, P. De Causmaecker and Y. Moreau. A guide to web tools to prioritize candidate genes. *Brief Bioinform* **12** (1) (2011), 22–32.

[113] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the international congress of mathematicians*, volume 2, pages 523–531, 1975.

[114] John W Tukey. We Need Both Exploratory and Confirmatory. *The American Statistician*, 34(1):23–25, February 1980.

[115] Simon Urbanek and Martin Theus. iplots: high interaction graphics for r. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Citeseer, 2003.

[116] S. Van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.

[117] Vergoulis, Thanasis and Vlachos, Ioannis S. and Alexiou, Panagiotis and Georgakilas, George and Maragkakis, Manolis and Reczko, Martin and Gerangelos, Stefanos and Koziris, Nectarios and Dalamagas, Theodore and Hatzigeorgiou, Artemis G., TarBase 6.0: capturing the exponential growth of miRNA targets with experimental support, *Nucleic Acids Research* 2011. doi = 10.1093/nar/gkr1161.

[118] M. K. Warmuth, G. Ratsch, M. Mathieson, J. Liao, and Ch. Lemmon. Active learning in the drug discovery process. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, USA, 2001. MIT Press.

[119] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer Publishing Company, Incorporated, 2009.

[120] Hadley Wickham. Bin-summarise-smooth: a framework for visualising large data. Technical report, had.co.nz, 2013.

[121] L. Xiong, B. Póczos, and J. Schneider. Hierarchical probabilistic models for group anomaly detection. *International Conference on Artificial Intelligence and Statistics*, 15:789–797, 2011.