18. Probabilistički grafički modeli II

Strojno učenje 1, UNIZG FER, ak. god. 2021./2022.

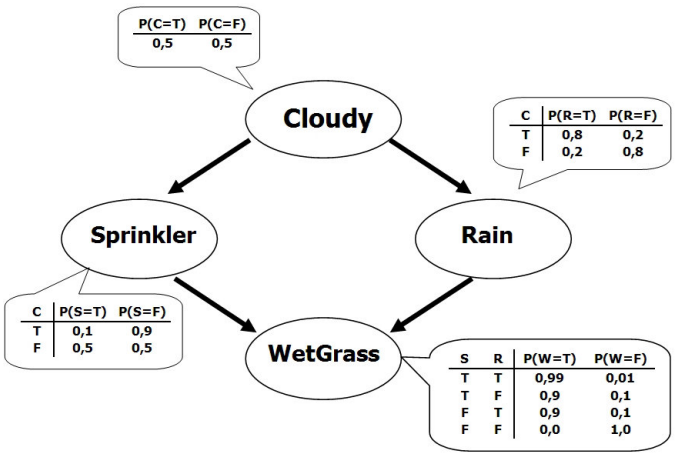
Jan Šnajder, predavanja, v2.0

Prošli smo puta pričali općenito o **probabilističkim grafičkim modelima** (PGM-ovima). Rekli smo da su to modeli koji na sažet način prikazuju zajedničku distribuciju, preko faktorizacije nad strukturom grafa. Rekli smo i da su tri aspekta PGM-ova **reprezentacija**, **zaključivanje** i **učenje**. Zatim smo se fokusirali na Bayesove mreže i pričali o reprezentaciji kod Bayesove mreže. Vidjeli smo kako Bayesova mreža kodira uvjetne nezavisnosti, vidjeli smo neke primjere Bayesovih mreža i zatim smo vidjeli kako možemo ispitati uvjetnu nezavisnost para varijabli pomoću metode d-odvajanja, što je pogotovo važno ako Bayesovom mrežom želimo modelirati i analizirati kauzalne odnose. 1

Danas nastavljamo pričati o Bayesovim mrežama, i to o njihova druga dva aspekta: zaključivanju i učenju. Pored toga, vratit ćemo se na kraju na širu sliku PGM-ova i dati neke napomene i smjernice. Kao što smo bili napomenuli prošli put, PGM-ovi su ogromno područje unutar strojnog učenja, i nemamo naravno ambiciju sve pokriti ovim predavanjem. U neke ćemo stvari ući malo više, no većina toga bit će ipak na razini informacije, u nadi da oni koje to zanima mogu dalje više informacija potražiti sami.

1 Zaključivanje

Rekli smo već da je najčešći zadatak koji želimo raditi s probabilističkim grafičkim modelima, a tako i s Bayesovom mrežom, **probabilističko zaključivanje**. Prisjetimo se primjera Bayesove mreže s prskalicom za travu i kišom.



Na primjer, ako opažamo da je trava mokra ($w = 1$), što možemo zaključiti o prskalici (s) i kiši (r) – što je od od tog dvoje vjerojatnije? To, zapravo, znači da nas zanimaju **uvjetne vjerojatnosti** $P(s = 1|w = 1)$ i $P(r = 1|w = 1)$. A njih, po definiciji uvjetne vjerojatnosti,

možemo izračunati na sljedeći način:

$$P(s=1|w=1) = \frac{P(s=1,w=1)}{P(w=1)}$$
$$P(r=1|w=1) = \frac{P(r=1,w=1)}{P(w=1)}$$

Vjerojatnost koja se pojavljuje u nazivniku, $P(w=1)$, naziva se **vjerojatnost dokaza** (u našem slučaju dokaz je činjenica da je trava mokra).

U ovim se izrazima javlja **zajednička vjerojatnost** dvije varijable (u brojniku razlomka) i marginalna vjerojatnost (u nazivniku razlomka). Sve te vjerojatnosti možemo izračunati iz Bayesove mreže. Prisjetimo se, naime, da Bayesova mreža upravo služi tome da na sažet način kodira zajedničku vjerojatnost. Konkretno, u ovom našem primjeru imamo (iščitavanjem iz Bayesove mreže):

$$P(c,s,r,w) = P(c)P(s|c)P(r|c)P(w|s,r)$$

Nadalje, prisjetimo se da iz ove zajedničke vjerojatnosti možemo izračunati bilo koju drugu vjerojatnost s ove četiri varijable, tako da napravimo prikladnu **marginalizaciju**. Konkretno, kako bismo izračunali tražene dvije uvjetne vjerojatnosti, zajedničku vjerojatnost $P(c,s,r,w)$ marginalizirat ćemo u brojniku i nazivniku izraza za uvjetnu vjerojatnost na sljedeći način:

$$P(s=1|w=1) = \frac{P(s=1,w=1)}{P(w=1)} = \frac{\sum_{c,r} P(c,s=1,r,w=1)}{\sum_{c,r,s} P(c,s,r,w=1)}$$
$$P(r=1|w=1) = \frac{P(r=1,w=1)}{P(w=1)} = \frac{\sum_{c,s} P(c,s,r=1,w=1)}{\sum_{c,r,s} P(c,s,r,w=1)}$$

Iščitavanjem konkretnih vrijednosti ovih vjerojatnosti iz Bayesove mreže i uvrštavanjem u gornje formule, dobivamo:

$$P(w=1) = \sum_{c,r,s} P(c,s,r,w=1) = 0.6471$$
$$P(s=1|w=1) = \frac{\sum_{c,r} P(c,s=1,r,w=1)}{\sum_{c,r,s} P(c,s,r,w=1)} = \cdots = 0.2781/0.6471 = 0.43$$
$$P(r=1|w=1) = \frac{\sum_{c,s} P(c,s,r=1,w=1)}{\sum_{c,r,s} P(c,s,r,w=1)} = \cdots = 0.4851/0.6471 = 0.708$$

Možemo zaključiti da je, ako opažamo mokru travu, vjerojatnije da pada kiša nego da je upaljena prskalica. Uključivanjem pozadinskog znanja o kauzalnim odmosima mogli bismo zaključiti da je vjerojatnije da je trava mokra zbog kiše nego zbog prskalice. Također smo izračunali da je vjerojatnost dokaza jednaka otprilike 0.6, što znači da mokru travu opažamo u 60% slučajeva.

Iz ovog smo primjera vidjeli da sve vjerojatnosti koja nas zanimaju možemo izračunati iz zajedničke vjerojatnosti prikladnom marginalizacijom i uvjetovanjem na opažane varijable. U tom smislu možemo reći da zajednička distribucija sadrži potpunu informaciju, iz koje se mogu derivirati sve druge distribucije koje nas zanimaju.

Općenito, **problem zaključivanja** kod PGM-ova može se definirati ovako: postoje varijable koje su vidljive odnosno **opažene** (engl. *observed*) i varijable koje nisu opažene odnosno koje su **skrivene** (engl. *hidden*). Npr., u našem primjeru, varijabla w bila je opažena, dok su sve druge varijable bile skrivene. Nadalje, od varijabli koje su skrivene, za neke nas baš zanima koja je njihova vjerojatnost, dok nas je za neke baš briga. Ove prve, koje nas zanimaju, nazivamo **varijable upita** (engl. *query variables*), a ove druge, koje nas ne zanimaju, nazivamo **varijable smetnje** (engl. *nuisance variables*). U našem primjeru, varijable upita bile su s (u prvom upitu) i r (u drugom upitu), dok su preostale varijable bile varijable smetnje.

Definirajmo ovo malo općenitije. Za neki upit, skup varijabli možemo razdijeliti u tri disjunktna skupa: **skup opaženih varijabli $\mathbf{x}_o$, skup varijabli upita $\mathbf{x}_q$ i skup varijabli smetnje $\mathbf{x}_n$**. Nakon što smo tako razdijelili varijable, možemo raditi različite upite. Dvije osnovne vrste upita su **aposteriorni upiti** i **MAP-upiti**.

Kod **aposteriornih upita**, zanima kakva je aposteriorna distribucija skrivenih varijabli za zadane opažene varijable. Dakle, neke varijable smo opazili, i sada nas zanima koja je vjerojatnost vrijednosti preostalih varijabli. To jest, zanima nas koliko iznosi sljedeća uvjetna vjerojatnost:

$$p(\mathbf{x}_q|\mathbf{x}_o) = \frac{\sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)}{p(\mathbf{x}_o)} = \frac{\sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)}{\sum_{\mathbf{x}_n, \mathbf{x}'_q} p(\mathbf{x}'_q, \mathbf{x}_o, \mathbf{x}_n)}$$

Primijetite kako smo u brojniku izmarginalizirali varijable smetnje $\mathbf{x}_n$. U nazivniku smo, uz varijable smetnje, izmarginalizirali i varijable upita $\mathbf{x}_q$, kako bismo dobili vjerojatnost dokaza. Vidimo da je ovo upravo ono što smo bili radili u našem ranijem primjeru. Pritom je u prvome upitu s bila varijabla upita, w je bila opažena varijabla, a c i r su bile varijable smetnje. U drugom upitu, r je bila varijabla upita, w je opet bila opažena varijabla, a c i s varijable smetnje.

Druga vrsta upita koja bi nas mogla zanimati jesu **MAP-upiti**, tj. upiti **maksimum aposteriori**. MAP-upiti također se nazivaju **najvjerojatnije objašnjenje** (engl. *most probable explanation, MPE*). Kod MAP-upita, zanima nas koje su najvjerojatnije vrijednosti za sve varijable upita, uz dane opažene varijable. Formalno:

$$\mathbf{x}_q^* = \operatorname*{argmax}_{\mathbf{x}_q} \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$$

Dakle, kod ove vrste upita ne zanima nas distribucija vjerojatnosti varijabli upita, nego nas zanima najvjerojatnije pridjeljivanje vrijednosti tim varijablama. Primijetite da najvjerojatnija vrijednost za vektor $\mathbf{x}_q$ nije nužno ista ona kao i kombinacija zasebno najvjerojatnijih vrijednosti za svaku pojedinačnu varijablu iz $\mathbf{x}_q$, kao što pokazuje sljedeći primjer.

▸ **PRIMJER**

Najvjerojatniji vektor $\mathbf{x}$ ne mora biti isto kao i kombinacija zasebno najvjerojatnijih komponenti tog vektora. Npr., ako je zajednička vjerojatnost $P(x, y)$ ovakva:

	x_1	x_2
y_1	0.4	0.3
y_2	0	0.3

onda je vektor (x_1, y_1) najvjerojatnija kombinacija para varijabli x i y, premda vrijednost x_1 zasebno gledano nije najvjerojatnija ($P(x_1) = 0.4$, ali $P(x_2) = 0.6$).

U našem primjeru s prskalicom i travom, MAP-upit bi na primjer mogao glasiti: ako opažamo da je trava mokra i da ne pada kiša, što je najvjerojatnija vrijednost za par skrivenih varijabli (c, s), tj. što je aposteriorno najvjerojatnije stanje oblačnosti i prskalice, ako apriorno znamo (jer to opažamo) da je trava mokra i da kiša ne pada.

2 Zaključivanje: Eliminacija varijabli

Kao što smo vidjeli, u načelu se obje ove vrste upita mogu odgovoriti tako da se najprije izgradi zajednička distribucija $p(\mathbf{x}_q, \mathbf{x}_o, \mathbf{x}_n)$, a potom se ili marginalizira pa normira (aposteriorni upiti) ili se nađe kombinacija s najvećom vjerojatnošću (MAP-upiti).

Međutim, takav je pristup naivan. Što je problem s takvim pristupom? Problem je što će izgradnja pune zajedničke distribucije pa njezina marginalizacija neminovno dovesti do **kombinatorne eksplozije**. Naime, u realnim je problemima broj varijabli vrlo velik, pa postoji mnogo

kombinacija njihovih vrijednosti. Općenito, ako je distribucija diskretna i n-dimenzijska, a svaka varijabla ima K mogućih vrijednosti, imat ćemo ukupno K^n kombinacija, tj. mogućih vrijednosti n-dimenzijskoga slučajnog vektora. Marginalizacija zajedničke distribucije iziskivat će onda množenje (da dobijemo zajedničku vjerojatnost) i zbrajanje (da provedemo marginalizaciju) velikog broja faktora, što će biti računalno zahtjevno. Zapravo, kada bismo to tako radili, u potpunost bismo poništili korist koju imamo od Bayesovih mreža – sažet zapis zajedničke distribucije!

Budući da je broj kombinacija eksponencijalan, pokazuje se da je zaključivanje kod Bayesovih mreža **NP-težak** problem. Međutim, postoje mudriji pristupi koji izbjegavaju eksplicitnu konstrukciju zajedničke vjerojatnosti. Alternative dolaze u dva okusa: **egzaktno zaključivanje** i **približno zaključivanje**.

Osnovni postupak egzaktnog zaključivanja je tzv. **eliminacija varijabli**. Eliminacija varijabli koristi **dinamičko programiranje**. Ideja je da se, umjesto da se prvo konstruira zajednička distribucija pa da se marginalizira (odnosno da se radi "izvana prema unutra"), da se ona gradi postepeno i da se pohranjuju međurezultati (to je onda "iznutra prema van"). Ovo je moguće zato što varijable u Bayesovoj mreži imaju lokalne zavisnosti: većina varijabli zavisi o manjem broju varijabli koje odgovaraju susjednim čvorovima. To znači da, kada računamo zajedničku vjerojatnost na temelju Bayesove mreže, mnogi će se izrazi ponavljati, pa je dovoljno da ih izračunamo samo jednom i pohranimo (memoiziramo) tu izračunatu vrijednost kako je poslije ne bismo morali ponovo računati. 2

▸ **PRIMJER**

Pogledajmo kako bi egzaktno zaključivanje izgledalo na našem primjeru s prskalicom za travu. Recimo da želimo izračunati distribuciju $p(w)$, dakle vjerojatnosti $P(w = 0)$ i $P(w = 1)$. (Iz didaktičkog razloga, zanemarimo sada činjenicu da je dovoljno da izračunamo jedno od tog dvoje pa da znamo ono drugo. Primjer bi imao više smisla kada bi varijabla w imala više od dvije vrijednosti.) Na temelju naše Bayesove mreže, vjerojatnost $P(w)$ možemo računati ovako:

$$\begin{aligned} P(w) &= \sum_c \sum_s \sum_r P(c, s, r, w) \\ &= \sum_c \sum_s \sum_r P(c)P(s|c)P(r|c)P(w|s, r) \end{aligned}$$

Problem kada radimo ovako jest u složenosti izračuna, odnosno u broju računskih operacija. Koliko računskih operacija ovdje imamo? Prvo, prisjetimo se da su sve varijable ovdje binarne. Ove tri sume generirat će, dakle, $2 \times 2 \times 2 = 8$ pribrojnika. Nadalje, svaki je pribrojnik umnožak od četiri faktora. Dakle, računanje svakog pribrojnika iziskuje 3 operacije množenja, i to ponavljamo 8 puta, dakle imamo 24 operacije množenja. Zatim zbrajamo te pribrojnike, što je još 7 operacija zbrajanja, pa je ukupan broj operacija 31. Na koncu to još trebamo ponoviti dva puta, za $w = 0$ i $w = 1$, što dakle daje ukupno 62 operacije.

Izračun postaje jednostavniji ako izlučimo zajedničke faktore. Na primjer:

$$\begin{aligned} P(w) &= \sum_c \sum_s \sum_r P(c, s, r, w) \\ &= \sum_c \sum_s \sum_r P(c)P(s|c)P(r|c)P(w|s, r) \\ &= \sum_c P(c) \sum_s P(s|c) \sum_r P(r|c)P(w|s, r) \end{aligned}$$

Iz jednostavnog razloga što se izlučeni članovi javljaju u izrazu samo jednom, to dovodi do smanjenja broja operacija koje moramo izračunati. Konkretno, ovdje bi to bile 42 operacije. Međutim, na ovaj način nismo ostvarili uštedu koju možemo ostvariti ako pohranjujemo izračune koji se ponavljaju. U gornjem izrazu niti jedan se izračun zapravo ne ponavlja.

Veću uštedu u broju izračuna možemo napraviti ako najprije presložimo redoslijed fakora, na

sljedeći način:

$$\begin{aligned} P(w) &= \sum_s \sum_r \sum_c P(c,s,r,w) \\ &= \sum_s \sum_r P(w|s,r) \sum_c P(c)P(s|c)P(r|c) \end{aligned}$$

Prednost ovakvog izračuna jest u tome što izraz unutar $\sum_c$ ne ovisi o varijabli c (jer po njoj marginaliziramo) ni o varijabli w (zbog lokalnosti, odnosno toga što faktori unutar sume ne ovise o varijabli w). Posljednja suma je, dakle, funkcija varijabli s i r. Uvedimo eksplicitno tu funkciju, i nazovimo ju $t_1(s,r)$. Dakle, uz $t_1(s,r) = \sum_c P(c)P(s|c)P(r|c)$, imamo:

$$P(w) = \sum_s \sum_r P(w|s,r)t(s,r)$$

Slično, marginalizacija varijabli r eliminira tu varijablu. Izraz koji je preostao ovisi samo o varijablama s i w, pa možemo uvesti funkciju $t_2(s,w) = \sum_r P(w|c,r)t_1(s,r)$. Onda imamo:

$$P(w) = \sum_s t_2(s,w)$$

Konačno, možemo uvesti funkciju $t_3(w) = \sum_s t_2(s,w)$, pa onda:

$$P(w) = t_3(w)$$

Izračunavanje $P(w)$ sada se svodi na izračun sljedećih funkcije za $w = 0$ i $w = 1$:

$$\begin{aligned} t_3(w) &= \sum_s t_2(s,w) \\ t_2(s,w) &= \sum_r P(w|c,r)t_1(s,r) \\ t_1(s,r) &= \sum_c P(c)P(s|c)P(r|c) \end{aligned}$$

Uštedu ovdje ostvarujemo pri izračunu funkcije $t_1(s,r)$, koja ne ovisi o w, pa je dakle dovoljno da je izračunamo samo jednom (za $w = 0$), pohranimo rezultat te ga iskoristimo idući put (za $w = 1$). Pogledajmo broj računskih operacija. Za izračun funkcije t_1 trebamo 5 operacija. Za izračun funkcije t_2 treba nam 8 opercija (od toga 5 za izračun funkcije t_1, ali njega ćemo napraviti samo jednom). Konačno, za izračun funkcije t_3 treba nam 17 operacija (od toga dva puta po 8 za dva izračuna funkcije t_2). Za izračun $P(w)$ ovo trebamo ponoviti za $w = 0$ i $w = 1$, pa je dakle ukupan broj operacija 34. To je manje nego 42 operacije, koliko smo imali u prethodnom slučaju.

Ovaj konceptualno jednostavan postupak zaključivanja naziva se **eliminacija varijabli zbroj-umnožak** (engl. *sum-product variable elimination*). Glavna je ideja da varijable marginaliziramo jednu po jednu, i svaki puta marginalizaciju radimo nad produktom faktora koji sadržavaju tu varijablu. Učinkovitost koja se pritom ostvaruje ovisi o konkretnoj strukturi Bayesove mreže. Što je manje bridova u mreži, to su varijable manje zavisne i faktori su to "lokalniji", pa je to više izraza koji se ponavljaju. 3

Inače, specifično za skrivene Markovljeve modele (HMM), algoritam eliminacije varijabli naziva se **algoritam unaprijed-unazad** (engl. *forward-backward algorithm*). Varijanta tog algoritma specifično za MAP-upite za HMM naziva se **Viterbijev algoritam**, i taj se algoritam vrlo često koristi u praksi (npr., u raspoznavanju govora i obradi prirodnog jezika).

Nažalost, za općenite grafove egzaktno je zaključivanje presloženo i traje eksponencijalno u broju čvorova (tj. varijabli). Konkretno, algoritamska složenost eliminacije varijabli izravno ovisi o veličini najvećeg faktora (jer po njemu trebamo obaviti sumaciju). U takvim slučajevima, alternativa egzaktnom zaključivanju je **približno zaključivanje** (engl. *approximate inference*). Tu

postoje dvije glavne grupe algoritama. Prva se grupa temelji na ideji da se konstruira neka jednostavna distribucija koja što bolje aproksimira ciljanu distribuciju, i zatim da se optimizira sličnost između tih dviju distribucija. Takvi postupci uključuju **propagacijske algoritme** (engl. *belief propagation*) i **varijacijsko zaključivanje** (engl. *variational inference*). Nažalost, nemamo vremena ulaziti u ove metode. Drugu grupu algoritama za približno zaključivanje čine algoritmi temeljeni na **uzorkovanju**. U nastavku ćemo malo pogledati neke od tih metoda.

3 Zaključivanje: Metode uzorkovanja

Metoda uzorkovanja metoda je, dakle, za približno zaključivanje. Prisjetimo se da kod zaključivanja mi zapravo želimo izračunati vjerojatnost varijabli upita, gdje su neke varijable možda već opažene, a neke nas ne zanimaju i to su varijable smetnje. Sad je pitanje kakve to ima veze s uzorkovanjem? Kako pomoću uzorkovanja možemo izračunati neku vjerojatnost? Zapravo, vrlo jednostavno. Ideja je da jednostavno uzorkujemo varijablu $\mathbf{x}$ iz njezine distribucije $P(\mathbf{x})$, tj. $\mathbf{x} \sim P(\mathbf{x})$, i to ponavljamo više puta kako bismo dobili uzorak vrijednosti te varijable iz dotične distribucije. Jednom kada imamo takav uzorak veličine N, možemo jednostavno izračunati očekivanje bilo koje vrijednosti varijable $\mathbf{x}$ kao relativnu frekvenciju te vrijednosti u uzorku:

$$P(\mathbf{x} = x) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\mathbf{x} = x\}$$

tj. zapravo računamo MLE procjenu kategoričke varijable. Zapravo, općenitije, možemo procijeniti očekivanu vrijednost bilo koje funkcije slučajne varijable $\mathbf{x}$:

$$\mathbb{E}[f] = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x})$$

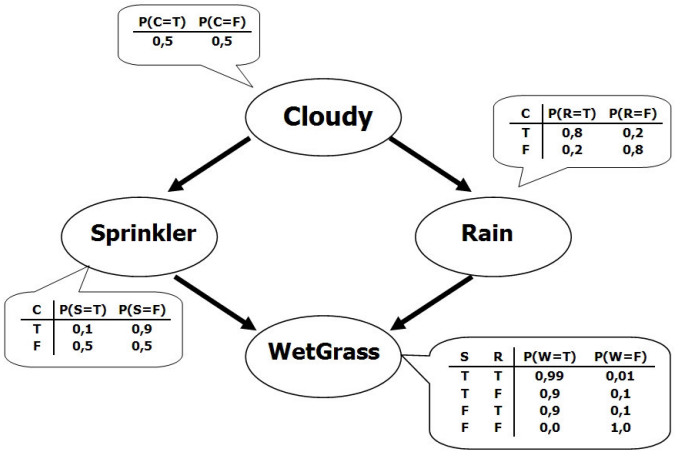
Na koncu, prisjetite se da baš na taj način procjenjujemo empirijsku pogrešku kao očekivanje funkcije gubitka.

To je, dakle, osnovna ideja – uzorkovati varijablu $\mathbf{x}$ iz distribucije $P(\mathbf{x})$, a onda na tom uzorku procijeniti parametar od interesa. U našem slučaju $P(\mathbf{x})$ je definirana pomoću Bayesove mreže. Sada se postavlja pitanje: kako uzorkovati iz distribucije prikazane Bayesovom mrežom?

3.1 Unaprijedno uzorkovanje

Prva ideja koja odmah svima padne na pamet jest da jednostavno uzorkujemo po strukturi Bayesove mreže, krenuvši od roditeljskih čvorova prema čvorovima djeci. Konkretno, krenemo od prvog čvora u mreži (prvog po topološkom uređaju) i slučajno generiramo, sukladno distribuciji tog čvora, jednu vrijednost za tu varijablu. Zatim idemo na iduću varijablu po topološkom uređaju i generiramo slučajnu vrijednost za tu varijablu. Ako je ta varijabla uvjetovana prvom varijablom, onda naravno uzimamo to u obzir kod generiranja, tj. kod čvorova djece generiramo iz uvjetne distribucije s pravilno postavljenim vrijednostima varijabli u čvorovima roditelja. Budući da idemo topološkim uređajem, zajamčeno je da ćemo u svakom čvoru do kojeg dođemo već imati generirane vrijednosti za sve varijable roditelja. Naposlijetku, kada tako obiđemo cijelu mrežu, imat ćemo jedan slučajan vektor iz zajedničke distribucije. Ponavljanjem ovog postupka generiramo niz ovakvih vektora, i oni onda čine naš uzorak. Ovaj se postupak naziva **unaprijedno uzorkovanje** (engl. *forward sampling*). Pogledajno unaprijedno uzorkovanje na primjeru prskalice i trave.

▸ **PRIMJER**



U prvom čvoru ("cloudy") uzorkovali bismo varijablu c iz distribucije za tu varijablu. Vjerojatnost da $c = 1$ je 0.5, ista kao i vjerojatnost za $c = 0$. Recimo da smo uzorkovali $c = 1$. U drugom koraku uzorkujemo iz čvora "sprinkler" iz uvjetne distribucije $p(s|c = 1)$. Tu je veća vjerojatnost da uzorkujemo $s = 0$, pa recimo da smo tako i dobili. U trećem koraku uzorkujemo u čvoru "rain" iz uvjetne distribucije $p(r|c = 1)$, i recimo da smo tu dobili $r = 1$. Konačno, u četvrtom koraku, kada već imamo $c = 1$, $s = 0$ i $r = 1$, uzorkujemo varijablu w u čvoru "wet grass" iz uvjetne distribucije $p(w|s = 0, r = 1)$. Recimo da smo tu dobili $w = 1$, jer je to vjerojatnije nego $w = 0$. Naš konačni slučajni vektor je dakle $\mathbf{x} = (c = 1, s = 0, r = 1, w = 1)$. Ponavljanjem ovog postupka dobili bismo uzorak vektora iz zajedničke distribucije $P(c, s, r, w)$.

3.2 Uzorkovanje s odbacivanjem

Unaprijedno uzorkovanje je superjednostavno, ali postoji problem. Nas u principu ne zanima uzorkovanje iz zajedničke distribucije, nego iz uvjetne distribucije! Naime, kada radimo upite nad mrežom, zanima nas aposteriorna vjerojatnost varijabli upita uz opažene varijable:

$$P(\mathbf{x}_q|\mathbf{x}_o)$$

Dakle, sada je pitanje kako možemo **uzorkovati iz uvjetne vjerojatnosti**? Bismo li za to mogli upotrijebiti postupak unaprijednog uzorkovanja? Problem je u tome što nekako moramo osigurati da su opažene varijable $\mathbf{x}_o$ postavljene na željene vrijednosti. Jedna mogućnost je da uzorkujemo kao i ranije, a onda da, iz uzorka koje dobijemo, uzmemo samo one vektore kod kojih su varijable $\mathbf{x}_o$ postavljene na točne vrijednosti, a sve ostale da odbacimo. Takav se postupak naziva **uzorkovanje s odbacivanjem** (engl. *rejection sampling*).

Vidite li problem s ovakvim pristupom? Problem je što je ćemo možda dobiti vrlo malo iskoristivih vektora. To će pogotovo biti slučaj kada je vjerojatnost dokaza $P(\mathbf{x}_o)$ jako malena. Tada ćemo, naime, vjerojatno zeznuti već kod uzorkovanja u početnim čvorovima. Npr., u našem primjeru, ako je $\mathbf{x}_o = (c = 1, s = 1)$, vjerojatnost da uzorkujemo takvu kombinaciju je samo $0.5 \cdot 0.1 = 0.05$. U složenijim Bayesovim mrežama lako se može dogoditi da za neke kombinacije vrijednosti varijabli jednostavno ne uspijevamo skupiti uzorak dovoljne veličine za pouzdanu procjenu.

3.3 Uzorkovanje po važnosti

Alternativa koja nam ovdje može pasti na pamet je da lijepo fiksiramo vrijednosti opažanih varijabli na željene vrijednosti, pa da onda uzorkujemo iz svih preostalih varijabli. Npr., ako je

$w = 1$, idemo tako fiskirati vrijednost varijable w, a onda ćemo uzorkovati iz preostalih varijabli. Međutim, tu je problem da ćemo tako dobiti uzorke koji ne poštuju pravu aposteriornu vjerojatnost (točnije: nećemo dobiti uzorke iz aposteriorne distribucije iz koje to želimo). Naime, ako fiksiramo $w = 1$, može nam se dogoditi da uzorkujemo $s = 0$ i $r = 0$, međutim prema tablici uvjetne vjerojatnosti u čvoru "wet" imamo $P(w = 1|s = 0, r = 0) = 0$, pa se to dakle uopće ne bi smjelo dogoditi. Slično, ako uzorkujemo uz opažanje $s = 1$, pa u vektoru postavimo $s = 1$, ali međutim nezavisno uzorkujemo c, onda možemo očekivati da ćemo dobiti 50% uzoraka sa $c = 0$ i 50% uzoraka sa $c = 1$, neovisno o tome što je $s = 1$. Međutim, to ne odgovara distribuciji definiranoj mrežom. Za tu distribuciju vidimo iz tablice uvjetnih vjerojatnosti da je pet puta manje vjerojatno da je $s = 1$ ako je $c = 1$ nego ako je $c = 0$.

Ovo bismo mogli korigirati, tako da damo težine uzorcima. Na primjer, ako fiksiramo $s = 1$, onda ćemo reći da su uzorci za koje $c = 1$ pet puta manje težine nego ovi za $c = 0$, budući da za izglednosti varijable c imamo $p(s = 1|c = 1) = 0.1$ i $p(s = 1|c = 0) = 0.5$. I doista, možemo to tako napraviti: možemo fiksirati vrijednosti svim opaženim varijablama, napraviti unaprijedno uzorkovanje, izračunati ovakve težine u svim čvorovima u kojima imamo opažene varijable, i onda pri izračunu očekivanja koristiti te težine. Takav postupak naziva se **uzorkovanje otežano izglednostima** (engl. *likelihood weighted sampling*), i jedan je od postupaka iz šire porodice postupka koji se nazivaju **uzorkovanje po važnosti** (engl. *importance sampling*).

Ovaj se postupak čini sasvim razumnim. Međutim, kada bismo analizirali njegova matematička svojstva (što ne budemo radili), ispada da postupak nije uvijek tako dobar. Konkretno, zanima nas kvaliteta procjenitelja (jer mi zapravo procjenjujemo parametre distribucije iz uzorka). Pokazuje se da kvaliteta ovisi o tome koliko je distribucija iz koje uzorkujemo (ona koja odgovara unaprijednom uzorkovanju iz Bayesove mreže uz fiksirane vrijednosti opaženih varijabli) različita od aposteriorne distribucije. Ako su te dvije distribucije vrlo slične, onda nema problema i naša će procjena biti dosta dobra. Međutim, ako su te dvije distribucije dosta različite, onda procjena neće biti dobra. Primijetimo da će distribucija iz koje uzorkujemo i aposteriona distribucija biti potpuno iste (i da nema potrebe za težinama) ako se sve opažene varijable nalaze u početnim čvorovima mreže (po topološkom uređaju), tj. ako su svi roditelji opaženih varijabli također opaženi. Tada, naime, zapravo uzorkujemo direktno iz aposteriorne distribucije. Suprotno tome, ako su sve opažene varijable u listovima, mi zapravo uzorkujemo iz apriorne distribucije, i ona može biti poprilično različita od aposteriorne distribucije. Tada se moramo osloniti na težine kako bismo dobili uzorak što sličniji onome kao da smo doista uzorkovai iz aposteriorne distribucije. Nažalost, to onda ipak smanjuje kvalitetu procjene.

3.4 Gibbsovo uzorkovanje

Uzorkovanje po važnosti je dobar postupak, ali možemo mi i bolje. Alternativu predstavljaju tzv. **Monte Carlo metode s Markovljevim lancem** (engl. *Markov chain Monte Carlo, MCMC*). Te metode uzorak generiraju slučajnim procesom, koji čini **Markovljev lanac**: niz slučajnih varijabli kod kojega iduća varijabla ovisi samo o trenutačnoj varijabli, ali ne i prethodnim varijablama u lancu. Lanac se konstruira tako da konvergira k stacionarnoj distribuciji koja je upravo ona distribucija iz koje želimo uzorkovati. Najpoznatije varijante algoritma MCMC su **Metropolis-Hastings** i **Gibbsovo uzorkovanje** (engl. *Gibbs sampling*). U nastavku ćemo ukratko opisati Gibbsovo uzorkovanje, koje se u praksi najviše koristi (također i za bayesovske modele, ali njih nećemo raditi).

Ideja Gibbsovog uzorkovanja je ova: krenemo od nekog slučajnog početnog vektora iz zajedničke distribucije, a onda ciklički uzorkujemo varijablu po varijablu tog vektora, tako da su uvijek sve varijable osim jedne fiksirane. Dakle, uvijek uzrokujemo samo jednu varijablu, iz distribucije koja je uvjetovana vrijednostima svih preostalih varijabli. Pogledajmo primjer.

▸ **PRIMJER**

Želimo uzorkovati iz zajedničke distribucije $p(x_1, x_2, x_3)$. Krenemo sa slučajnim vektorom $\mathbf{x}^0$ iz uzorkovanim iz distribucije $p(x_1^0, x_2^0, x_3^0)$, npr., unaprijednim uzorkovanjem (broj u eksponentu označava broj iteracije). Zatim uzrokujemo varijable redom, i svaki puta prethodnu vrijednost te varijable nadomjestimo vrijednošću koju smo upravo dobili uzorkovanjem. Konkretno, najprije uzorkujemo x_1^1 iz uvjetne distribucije $p(x_1|x_2^0, x_3^0)$. Dobivamo x_1^1. Zatim uzorkujemo x_2^1 iz uvjetne distribucije $p(x_2|x_1^1, x_3^0)$ te dobivamo x_2^1. Zatim uzorkujemo x_3^1 iz uvjetne distribucije $p(x_3|x_1^1, x_2^1)$. Nakon ova tri koraka, završili smo prvu iteraciju uzorkovanja te imamo naš prvi slučajni vektor, $\mathbf{x}^1 = (x_1^1, x_2^1, x_3^1)$. Sada bismo ovaj postupak ponavljali i generirali daljne slučajne vektore. Dakle, postupak uzorkovanja je ovaj:

$$
\begin{aligned}
&\mathbf{x}^0 \sim p(x_1^0, x_2^0, x_3^0) && \Rightarrow \text{prvi vektor (npr., unaprijednim uzorkovanjem)} \\
&x_1^1 \sim p(x_1|x_2^0, x_3^0) \\
&x_2^1 \sim p(x_2|x_1^1, x_3^0) \\
&x_3^1 \sim p(x_3|x_1^1, x_2^1) && \Rightarrow \text{vektor } \mathbf{x}^1 = (x_1^1, x_2^1, x_3^1) \\
&x_1^2 \sim p(x_1|x_2^1, x_3^1) \\
&x_2^2 \sim p(x_2|x_1^2, x_3^1) \\
&x_3^2 \sim p(x_3|x_1^2, x_2^2) && \Rightarrow \text{vektor } \mathbf{x}^2 = (x_1^2, x_2^2, x_3^2) \\
&\vdots
\end{aligned}
$$

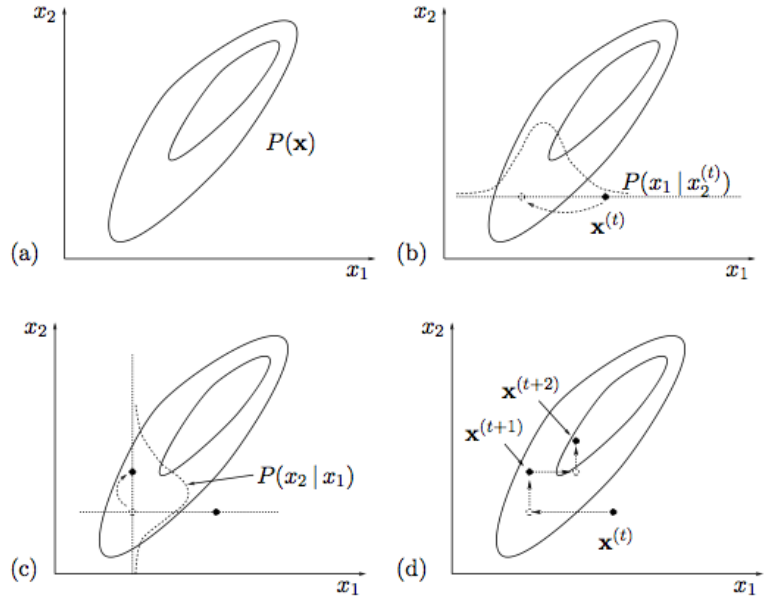
Budući da Gibbosovo uzorkovanje započinjemo od nekog slučajnog vektora, koji može biti poprilično nereprezentativan (tj. malo vjerojatan) za distribuciju iz koje uzorkujemo, općenito treba proći određeni broj iteracija prije nego što uzorkovanje dođe do područja veće gustoće vjerojatnosti i vektori postanu reprezentativni za distribuciju. Da bismo to ostvarili, lanac uzorkovanja treba biti dovoljno dug. Isto tako se preporuča odbaciti određeni broj početno dobivenih vektora, dok se lanac ne "zavrti" u području veće gustoće vjerojatnosti, odnosno dok ne završi tzv. **period zagrijavanja** (engl. *burn-in period*).

Očito, Gibbsovo uzorkovanje pretpostavlja da možemo jednostavno izračunati uvjetnu distribuciju jedne varijable uvjetovane na sve ostale varijable te da možemo egzaktno uzorkovati iz te distribucije. Ovdje nam u prilogu ide činjenica da kod Bayesovih mreža svaka varijabla uvjetno ovisi tek o manjem broju drugih varijabli. Preciznije, svaka uvjetna distribucija neke varijable u Bayesovoj mreži ovisi samo o varijablama u **Markovljevom omotaču** (engl. *Markov blanket*) čvora koji odgovara toj varijabli. Markovljev omotač čine roditeljske čvorovi, čvorovi djece te roditelji čvorova djece. Zbog tako lokalizirane ovisnosti, općenito je jednostavno napisati distribuciju neke varijable uvjetovanu na sve ostale varijable. Ipak, budući da varijable u Bayesovoj mreži općenito mogu imati različite distribucije (ne moraju nužno biti kategoričke), izračun uvjetne distribucije, a time i egzaktno uzorkovanje iz takve distribucije, općenito mogu biti netrivijalni.

▸ **PRIMJER**

Pogledajmo Gibbsovo uzorkovanje iz bivarijatne Gaussove distribucije:

$$p(x_1, x_2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



Na slici (a) gore lijevo prikazane su izokonture bivarijatne Gaussove distribucije $p(x_1, x_2)$. Na slici (b) gore desno prikazan je prvi korak Gibbsovog uzorkovanja, gdje inicijalno krećemo od slučajnog vektora (točke) $\mathbf{x}^t$ i zatim uzorkujemo prvu varijablu x_1 iz uvjetne distribucije $p(x_1|x_2^{(t)})$. Ta uvjetna distribucija bivarijatne Gaussove distribucije je također Gaussova distribucija. Naime, svaka distribucija podskupa varijabli multivarijatne Gaussove distribucije uvjetovane na podskup preostalih varijabli je i sama je multivarijatna Gaussova distribucija, a u slučaju Gibbsovog uzorkovanja iz Gaussove distribucije to je uvijek univarijatna Gaussova dristribucija budući da su sve varijable osim jedne fiksirane. Dakle, varijablu x_1 uzorkujemo iz uvjetne distribucije (odnosno gustoće vjerojatnosti) $p(x_1|x_2^{(t)})$, i to je univarijatna Gaussova distribucija koja je na slici (b) prikazana iscrtkano. Recimo da smo uzorkovali vrijednost koja je naznačena na slici, pa smo napravili pomak ulijevo u smjeru osi $-x_1$, kako je indicirano strelicom. Zatim, u drugom koraku, uzorkujemo varijablu x_2 iz uvjetne distribucije (odnosno gustoće vjerojatnosti) $p(x_2|x_1)$, što je opet univarijatna Gaussova distribucija, te se pomičemo, npr., u smjeru osi $+x_2$ kako je prikazano na slici (c) dolje lijevo. Time smo dobili prvi vektor (točku) $\mathbf{x}^{t+1}$. Sada ponavljamo postupak. Kroz niz koraka, kao što ilustrira slika (d) dolje desno, generiramo niz točaki, većina kojih će biti smještena u području navjeće gustoće naše bivarijatne Gaussove distribucije, odnosno točke će biti distribuirane po toj distribuciji.

U gornja dva primjera uzorkovali smo iz zajedničke distribucije. Međutim, kao što smo rekli, tipično nas zanima uzorkovanje iz aposterione distribucije $p(\mathbf{x}_q|\mathbf{x}_o)$, gdje su neke varijable $\mathbf{x}_o$ već opažene. U tom slučaju, kod Gibbosovog uzorkovanja jednostavno ćemo te opažene varijable fiksirati na njihove opažene vrijednosti te nećemo uzorkovati vrijednosti za te varijable. Uzorkovanjem iz varijabli upita $\mathbf{x}_q$, ciklički kako je opisano gore, dobit ćemo upravo uzorak iz aposteriorne distribucije uvjetovane na opažene varijable $\mathbf{x}_o$.

▸ **PRIMJER**

Pogledajmo opet primjer s prskalicom za travu. Ako znamo da je trava mokra ($w = 1$), što možemo zaključiti o kiši (r)? Drugim riječima, zanima nas aposteriorna distribucija $P(r|w = 1)$. Budući da je varijabla w opažena s vrijednošću 1, fiksirat ćemo $w = 1$, dok ćemo vrijednosti ostalih varijabli uzorkovati. Krećemo od nekog slučajnog vektora, generiranog unaprijednim uzorkovanjem. Neka je to vektor $\mathbf{x}^0 = (c = 0, s = 0, r = 1, w = 1)$. U prvom koraku uzorkujemo varijablu c iz uvjetne distribucije $P(c|s = 0, r = 1, w = 1)$, zanemarujući inicijalnu vrijednost varijable c. Tu uvjetnu

distribuciju možemo lako izračunati na temelju naše Bayesove mreže:

$$P(c|s=0,r=1,w=1) = \frac{P(c,s=0,r=1,w=1)}{P(s=0,r=1,w=1)} = \frac{P(c)p(s=0|c)p(r=1|c)P(w=1|s=0,r=1)}{\sum_c P(c)P(s=0|c)P(r=1|c)P(w=1|s=0,r=1)}$$

Vrijednost za varijablu c uzorkovali bismo iz ove distribucije (u ovom slučaju to je Bernoullijeva distribucija jer je c binarna varijabla). U idućem koraku fiksiramo varijablu c na dobivenu vrijednost, dok ostale varijable ostavljamo na njihovim prethodnim vrijednostima, osim varijable s, koju bismo iduću uzorkovali. Zatim bismo isto napravili za varijablu r. Nakon toga, ciklus se ponavlja od varijable c, pa s, pa opet r, itd. Varijablu w ne bismo uzorkovali jer je ona opažena, i njezina je vrijednost fiksirana na $w = 1$. Na taj bismo način dobili uzorak iz distribucije $P(c, s, r|w = 1)$, te iz tog uzorka možemo procijeniti tu distribuciju.

Ovdje možda izgleda kao da nismo puno dobili Gibbsovim uzorkovanjem u odnosu na egzaktno zaključivanje, jer svaki puta moramo izračunati uvjetnu distribuciju iz koje uzorkujemo, a da bismo to napravili moramo množiti faktore i marginalizirati, baš kao i kod egzaktnog zaključivanja. Doista, kod ovako male Bayesov mreže egzaktno zaključivanje zapravo nije problematično. Međutim, kod većih mreža, koje reprezentiraju visokodimenzijske zajedničke distribucije, egzaktno je zaključivanje, kako smo već napomenuli, netraktabilno. Gibbsovo uzorkovanje onda predstavlja učinkovito alternativu. Naime, premda za Gibbsovo uzorkovanje moramo izračunati uvjetnu distribuciju za varijablu po kojoj uzorkujemo, ta će distribucija uvijek ovisiti o manjem broju susjednih varijabli u mreži (već spomenuti Markovljev omotač), pa dakle nemamo problem s netraktabilnošću kao kod egzaktnog zaključivanja.

Konačno, budući da nas zapravo zanima distribucija $P(r|w = 1)$, nju možemo izračunati iz distribucije $P(c, s, r|w = 1)$ tako da marginaliziramo po varijablama s i r (koje su u ovom upitu varijable smetnje):

$$\begin{aligned} P(r|w=1) &= \sum_s \sum_r P(c,s,r|w=1) \\ &= P(c=0,s=0,r|w=1) + P(c=0,s=1,r|w=1) + \\ &\quad P(c=1,s=0,r|w=1) + P(c=1,s=1,r|w=1) \end{aligned}$$

Svi ove četiri vjerojatnosti mogu se procijeniti iz našeg uzorka izvučenog iz $P(c, s, r|w = 1)$, pa dakle ovo nije problem izračunati.

4 Učenje

PGM-ovi su probabilistički modeli. Kako ih učimo, odnosno na što se svodi učenje kod probabilističkih modela? Učenje se svodi na **procjenu parametara $\boldsymbol{\theta}$** distribucije koja opisuje podatke. Kako ćemo izračunati procjenu parametara te distribucije? Kao i uvijek, na raspolaganju imamo tri alata: **MLE, MAP i bayesovsku procjenu** (ovu treću nećemo raditi).

Prošli put spomenuli smo **skriveni Markovljev model** (engl. *Hidden Markov Model, HMM*). Karakteristika tog modela bila je da sadrži varijable koje su skrivene, tj. koje **nisu opažene u podatcima**. Postoje mnogi drugi PGM-ovi koji imaju skrivene varijable, budući da uvođenje skrivenih varijabli može znatno pojednostaviti modeliranje. No, pokazuje se da se postupak učenja bitno razlikuje ovisno o tome ima li PGM skrivenih varijabli ili ne. Naime, ako model ima skrivenih varijabli, onda te varijable nikada nisu opažene u podatcima (te varijable postoje samo u modelu, ali ne i u podatcima), pa nećemo moći samo tako procijeniti parametre njihovih distribucija. U tom slučaju govorimo o učenju iz **nepotpunih podataka** (engl. *incomplete data*) – u podatcima nemamo sve varijable koje imamo u modelu. Suprotno, ako model nema skrivenih varijabli, onda to znači da sve varijable opažamo iz podataka, i u tom slučaju govorimo u učenju

iz **potpunih podataka** (engl. *complete date*). Učenje iz potpunih podataka je jednostavnije, pa ćemo njega prvog razmotriti.

4.1 Potpuni podatci

Najjednostavniji način procjene parametara je MLE. Znamo MLE procjenitelje za standardne teorijske distribucije. Međutim, Bayesova mreža odgovara nekoj složenijoj distribuciji, za čije parametre ne znamo kako glasi MLE procjenitelj. Kako ćemo onda izračunati MLE procjenu za parametre te distribucije? Napravit ćemo to kao i ranije: izvest ćemo MLE procjenitelj **maksimizacijom log-izglednosti**. Trebamo, dakle, napisati log-izglednost parametara distribucije koja odgovara Bayesovoj mreži, a to je jednostavno. Najprije, prisjetimo se, od prošlog puta, da je zajednička vjerojatnost $p(\mathbf{x})$ pomoću Bayesove mreže faktorizirana na sljedeći način:

$$p(\mathbf{x}) = \prod_{k=1}^{n} p(x_k|\text{pa}(x_k))$$

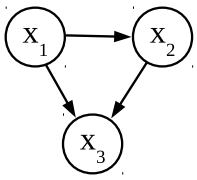
gdje je $\text{pa}(x_k)$ skup roditeljskih čvorova čvora x_k. Parametri ove distribucije neka su $\boldsymbol{\theta}$. Log-izglednost parametara $\boldsymbol{\theta}$ onda je:

$$\begin{aligned}
\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) &= \ln p(\mathcal{D}|\boldsymbol{\theta}) = \ln p(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}|\boldsymbol{\theta}) \\
&= \ln \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) = \ln \prod_{i=1}^{N} \prod_{k=1}^{n} p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k) \\
&= \ln \prod_{k=1}^{n} \prod_{i=1}^{N} p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k) = \sum_{k=1}^{n} \sum_{i=1}^{N} \ln p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k)
\end{aligned}$$

Ovdje primijetite da imamo dva produkta: jedan ide po svim primjerima ($i = 1, \ldots, N$), jer izglednost računamo kao umnožak vjerojatnosti za sve primjere iz skupa $\mathcal{D}$, a drugi ide po svim značajkama ($k = 1, \ldots, n$), jer za izračun zajedničke vjerojatnosti moramo pomnožiti sve faktore, a u Bayesovoj mreži imamo po jedan faktor za svaku značajku. Budući da produkti komutiraju, možemo zamijeniti njihov redosijed. Nakon što primijenimo logaritam, produkti se pretvaraju u sume, pa tako dobivamo sumu po faktorima (vanjska suma), gdje za svaki faktor sumiramo po primjerima (unutarnja suma). Kažemo da se log-izglednost **dekomponirala** prema strukturi grafa Bayesove mreže. Pod time mislimo da smo dobili po jedan pribrojnik za svaki čvor Bayesove mreže. Pogledajmo primjer.

▸ **PRIMJER**

Razmotrimo sljedeću Bayesovu mrežu:



Pripadna faktorizacija je:

$$p(x_1, x_2, x_3) = p(x1)p(x_2|x_1)p(x_3|x_1, x_2)$$

Log-izglednost parametara ove distribucije je:

$$\begin{aligned}\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) &= \ln p(\mathcal{D}|\boldsymbol{\theta}) \\ &= \sum_{k=1}^{n}\sum_{i=1}^{N} \ln p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k) \\ &= \sum_{i=1}^{N} \ln p(x_1^{(i)}|\text{pa}(x_1^{(i)}), \boldsymbol{\theta}_1) + \sum_{i=1}^{N} \ln p(x_2^{(i)}|\text{pa}(x_2^{(i)}), \boldsymbol{\theta}_2) + \sum_{i=1}^{N} \ln p(x_3^{(i)}|\text{pa}(x_3^{(i)}), \boldsymbol{\theta}_3) \\ &= \sum_{i=1}^{N} \ln p(x_1^{(i)}|\boldsymbol{\theta}_1) + \sum_{i=1}^{N} \ln p(x_2^{(i)}|x_1^{(i)}, \boldsymbol{\theta}_2) + \sum_{i=1}^{N} \ln p(x_3^{(i)}|x_1^{(i)}, x_2^{(i)}, \boldsymbol{\theta}_3)\end{aligned}$$

Dekompozicija log-izglednosti je dobra stvar, jer to znači da možemo za svaki čvor mreže nezavisno procijeniti parametre $\boldsymbol{\theta}_k$ uvjetne distribucije za varijablu x_k. Konkretno, MLE procjena parametara uvjetne distribucije za k-tu varijablu je:

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\text{argmax}} \sum_{i=1}^{N} \ln p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k)$$

Slično, MAP procjena parametra uvjetne distribucije za k-tu varijablu je:

$$\boldsymbol{\theta}_k^* = \underset{\boldsymbol{\theta}_k}{\text{argmax}} \Big(\sum_{i=1}^{N} \ln p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k) + \ln p(\boldsymbol{\theta}_k) \Big)$$

gdje je $p(\boldsymbol{\theta}_k)$ apriorna gustoća vjerojatnosti parametra $\boldsymbol{\theta}_k$. Prisjetimo se: ako želimo rješenje u zatvorenoj formi, onda $p(\boldsymbol{\theta}_k)$ treba biti konjugatna za izglednost $\sum_i p(x_k^{(i)}|\text{pa}(x_k^{(i)}), \boldsymbol{\theta}_k)$. Prisjetimo se, također, da ako je apriorna vjerojatnost parametara $p(\boldsymbol{\theta}_k)$ uniformna, MAP degenerira na MLE.

Naravno, sad je pitanje koja je to konkretno distribucija u pitanju, koje pojedini čvorovi modeliraju. Distribucija je najčešće diskretna tj. **kategorička** (ali ne mora biti, može npr. biti Gaussova). Kao što znamo, za kategoričku varijablu MLE procjena za μ_k je jednostavno relativna frekvencija za realizaciju vrijednosti svake vrijednosti. Malo preciznije, ako je k indeks varijable (čvora), j je vrijednost uvjetovanih varijabli (roditeljskih čvorova), a l je vrijednost varijable, onda:

$$N_{kjl} = \sum_{i=1}^{N} \mathbf{1}\{\mathbf{x}_{\text{pa}(x_k)}^{(i)} = j \wedge x_k^{(i)} = l\}$$

Broj primjera za koje je vrijednost varijabli roditelja čvora k jednaka j:

$$N_{kj} = \sum_{l} N_{kjl}$$

MLE procjena je onda relativna frekvencija:

$$\hat{\mu}_{k,j,l} = \frac{N_{k,j,l}}{N_{k,j}}$$

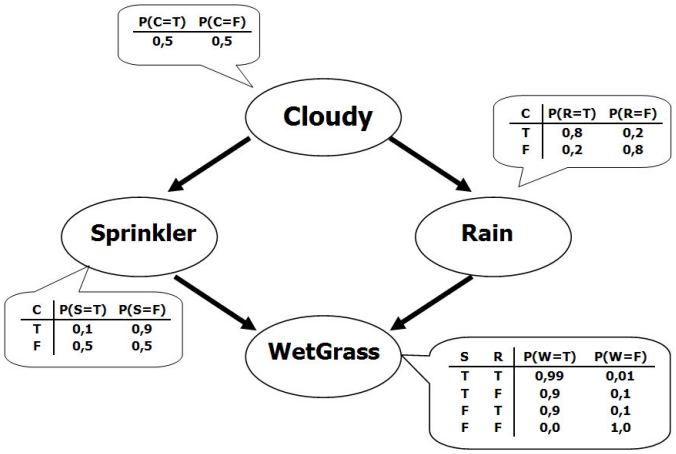
Ako želimo raditi MAP-procjenu onda (Dirichlet-kategorički model uz $\alpha = 2$):

$$\hat{\mu}_{k,j,l} = \frac{N_{kjl} + 1}{N_{kj} + K_k}$$

gdje je K_k broj mogućih vrijednosti varijable x_k.

▸ **PRIMJER**

Prisjetimo se Bayesove mreže s prskalicom za travu:



Recimo da želimo procijeniti, pomoću MAP, parametre za čvor w. Čvor w odgovara uvjetnoj vjerojatnosti $P(w|s,r)$. MAP procjena je:

$$\hat{\mu}_{w,(s,r),1} = P(w=1|s,r) = \frac{\sum_{i=1}^{N} \mathbf{1}\{x_s^{(i)} = s \wedge x_r^{(i)} = r \wedge x_w^{(i)} = w\} + 1}{\sum_{i=1}^{N} \mathbf{1}\{x_s^{(i)} = s \wedge x_r^{(i)} = r\} + 2}$$

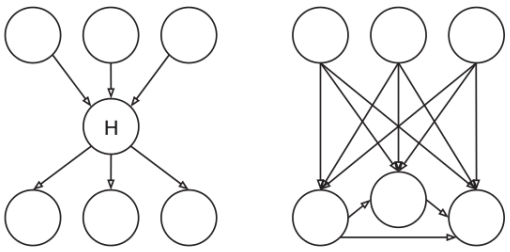
4.2 Nepotpuni podatci

Kako što smo već napomenuli, PGM možemo definirati tako da uključuje skrivene varijable. Tipično su to varijable koje smo uveli u model kao "posredničke varijable" između varijabli koje opažamo. Npr. ako modeliramo vezu između uzorka bolesti (npr. pušenje) i simptoma (npr. bol u prsima), skrivena varijabla koju ne opažamo je sama bolest (srčana bolest). Skrivene varijable nikada ne opažamo u podatcima (one u podatcima ne postoje), i zato u takvom slučaju govorimo u učenju iz **nepotpunih podataka**. 7

Sljedeći primjer ilustrira pogodnosti uvođenja skrivene varijable u model.

▸ **PRIMJER**

Razmotrimo dvije Bayesove mreže: 8



Obje Bayesove mreže opisuju vezu između primarnih uzroka bolesti (gornji čvorovi) i simptoma (donji čvorovi). Mreža na lijevoj slici sadrži skrivenu varijablu H (čvor u sredini) koji modelira bolest kao zajedničku posljedicu primarnih uzroka i zajednički uzrok svih simptoma. Ta je varijabla skrivena jer bolest, za razliku od primarnih uzroka i simptoma i, ne možemo izravno opažati. Desna slika prikazuje Bayesovu mrežu koja modelira iste odnose između primarnih uzroka bolesti i simptoma, ali

bez posredovanja skrivene varijable H. U ovom modelu imamo izravno povezane čvorove primarnih uzroka i simptoma, te dodatne veze između simptoma. Lijevi je model znatno intuitivniji jer uvažava postojanje posredničkog konstrukta (bolesti), unatoč tome što taj konstrukt ne možemo izravno opažati (općenito, skrivene varijable mogu biti, i zapravo često i jesu, nekakvi hipotetski i apstraktni konstrukti). No, osim konceptualne jednostavnosti, lijevi je model znatno jednostavniji u smislu broja parametara. Naime, uz pretpostavku da su sve varijable binarne, lijevi model ukupno ima 17 parametara, dok desni model ukupno ima 59 parametara (uvjerite se u to!). Čak i ako bismo u desnom modelu uklonili veze između simptoma – takav model bolje bi odgovarao lijevom modelu sa skrivenom varijablom – i dalje bi broj parametara bio veći nego kod modela sa skrivenom varijablom (imali bismo ukupno 27 parametara).

Ranije smo vidjeli da, kada učimo Bayesovu mrežu iz potpunih podataka, log-izglednost se dekoponira po strukturi Bayesove mreže, i to nam omogućava da parametre modela procijenimo zasebno za svaki čvor. Nadalje, ako su distribucije za pojedinačne varijable neke teorijske distribucije (iz eksponencijalne familije), onda MLE i MAP procjene dobivamo u zatvorenoj formi. Međutim, kod učenja modela sa skrivenim varijabla, tj. učenja iz nepotpunih podataka, log-izglednost se neće dekomponirati po čvorovima mreže, i to komplicira postupak učenja.

Pokažimo u čemu je problem. Mogli bismo to pokazati na bilo kojem modelu sa skrivenim varijablama, npr. skrivenom Markovljevom modelu. No, izabrat ćemo jedan jednostavniji model sa skrivenim varijablama: **mješavinski model** (engl. *mixture model*). O tom modelu ćemo pričati u kontekstu nenadziranog učenja, točnije grupiranja (klasteringa). Mješavinski modeli zajedničku vjerojatnost modeliraju na sljedeći način

$$P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = P(\mathbf{z}|\boldsymbol{\theta})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$$

Ovo na prvi pogled izgleda kao naivan Bayesov klasifikator (gdje bi $\mathbf{z}$ bila oznaka klase), no problem je što kod nenadziranog učenja $\mathbf{z}$ nije opažena varijabla, jer kod nenadziranog učenja ne znamo kojoj klasi primjer pripada. Dakle, varijable $\mathbf{z}$ nisu opažene varijable, tj. to nisu oznake $\mathbf{y}$. U tom smislu naši su podatci nepotpuni: raspolažemo samo opažanjima za varijablu $\mathbf{x}$, ali ne i opažanjima za skrivenu varijablu $\mathbf{z}$.

Pogledajmo kako bismo izrazili log-izglednost parametara $\boldsymbol{\theta}$ ovog modela:

$$\begin{aligned}
\ln \mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \ln p(\mathcal{D}|\boldsymbol{\theta}) &= \ln p(\mathbf{x}_1, \ldots, \mathbf{x}_N|\boldsymbol{\theta}) \\
&= \ln \prod_{i=1}^{N} \sum_{\mathbf{z}} p(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}|\boldsymbol{\theta}) \\
&= \ln \prod_{i=1}^{N} \sum_{\mathbf{z}} p(\mathbf{z}^{(i)}|\boldsymbol{\theta}) p(\mathbf{x}^{(i)}|\mathbf{z}^{(i)}, \boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \ln \sum_{\mathbf{z}} p(\mathbf{z}^{(i)}|\boldsymbol{\theta}) p(\mathbf{x}^{(i)}|\mathbf{z}^{(i)}, \boldsymbol{\theta})
\end{aligned}$$

Primijetite da smo proveli marginalizaciju po varijabli $\mathbf{z}$, jer tu varijablu imamo u modelu, ali je ne opažamo u skupu $\mathcal{D}$, pa je moramo marginalizirati kako bismo izrazili vjerojatnost skupa $\mathcal{D}$. No, kao što vidimo, za razliku od log-izglednosti potpunih podataka, ovdje logaritamsku funkciju ne možemo potisnuti unutar sume (logaritamska je funkcija "zapela" između dvije sume), pa se log-izglednost **ne dekomponira** po varijablama. Posljedica toga je da ne postoji rješenje u **zatvorenoj formi**, niti za MLE, MAP niti bayesovski procjenitelj. Umjesto toga, moramo koristiti neki iterativni optimizacijski algoritam. Jedna mogućnost je **gradijentni uspon** (engl. *gradient ascent*) (uspon, a ne spust, jer maksimiziramo log-izglednost, ali princip je identičan kao i kod gradijentnog spusta). Međutim, za mješavinske modele i skrivene Markovljeve modele češće se koristi **algoritam maksimizacije očekivanje (EM-algoritam)**

(engl. *expectation maximization algorithm*), o kojem ćemo pričati u kontekstu algoritama grupiranja. Ukratko, ideja EM-algoritma je sljedeća: kada bi sve varijable bile opažene, onda bismo lako izračunali MLE/MAP. Ideja je onda da izačunamo **očekivane vrijednosti** svih varijablu (zapravo, aposteriorne distribucije za sve varijable), koristeći postupak **zaključivanja**, a zatim te 9
očekivane vrijednosti koristimo kao da su opažene vrijednosti. Na temelju očekivanih vrijednosti varijabli zatim izračunamo MLE/MAP procjene, tj. provodimo maksimizaciju. Nakon toga, iznova izračunamo očekivane vrijednosti varijabli, i zatim taj postupak iterativno ponavljamo do konvergencije, alternirajući između koraka izračuna očekivanja i maksimizacije vjerojatnosti.

Osim što rješenje za MLE/MAP ne postoji u zatvorenoj formi, drugi problem s učenjem iz nepotpunih podataka jest što log-izglednost (za MLE) odnosno aposteriorna vjerojatnost (za MAP) više nisu konveksne funkcija parametara $\boldsymbol{\theta}$. Te funkcija niti su konveksne niti konkavne, 10
već imaju **lokalne maksimume**, što znači da optimizacija ne mora dati globalno optimalne parametre. EM-algoritam i gradijentni uspon konvergirat će u neki lokalni optimum funkcije izglednosti (ovisno o početnim uvjetima), pa je dobro da se postupak optimizacije više puta ponovi, sa slučajno odabranim početnim parametrima.

Sažetak

- Zaključivanje kod **probabilističkih grafičkih modela (PGM-ova)** svodi se na određivanje vjerojatnosti **varijabli upita** uvjetovane na **opažene varijable**, uz marginalizaciju **varijabli smetnje**
- Postoje **aposteriorni upiti** i **MAP-upiti**
- **Eliminacija varijabli** je egzaktan postupak zaključivanja koji tehnikom **dinamičkog programiranja** izbjegava problem kombinatorne eksplozije pri konstrukciji zajedničke distribucije
- Za općenite PGM-ove egzaktni su postupci presloženi, pa koristimo **približno zaključivanje**
- Kod **metoda uzorkovanja** približno zaključivanje ostvaruje se procjenom parametara iz uzorka dobivenog uzorkovanjem PGM-a
- **Unaprijedno uzorkovanje**, **uzorkovanje s odbacivanjem** i **uzorkovanje po važnosti** su metode uzorkovanja iz zajedničke odnosno aposteriorne distribucije definirane Bayesovom mrežom
- **Gibbsovo uzorkovanje** generira uzorke pomoću Markovljevog lanca, uzorkujući varijablu po varijablu
- **Učenje** Bayesove mreže svodi se na MLE ili MAP procjenu parametara svake varijable zasebno, jer se log-izglednost dekomponira po čvorovima mreže
- Ako model ima **skrivene varijable**, onda učimo iz **nepotpunih podataka**, za što moramo koristiti iterativne algoritme, npr. **gradijentni uspon** ili **algoritam maksimizacije očekivanja**

Bilješke

1 Ovo se predavanje dominantno oslanja na [Koller and Friedman, 2009] (poglavlja 9 i 12). Također preporučam [Bishop, 2006] (poglavlja 8.4 i 11.1–3) i [Murphy, 2012] (poglavlja 10.4, 20.3, 23.1–4, 24.1–2).

2 Vjerujem da vam je koncept **dinamičkog programiranja** poznat. Meni se svidjela knjižica [Denardo, 2012], koja daje iscrpan opis tog područja. Za kratak uvod, pogledajte `http://20bits.com/article/introduction-to-dynamic-programming`. Dobro je ne brkati **memoizaciju** (pohranjivanje jednom izračunatih rezultata) i dinamičko programiranje (način rješavanja problema koji iskorištava preklapanje podproblema, a tipično se ostvaruje rekurzijom s memoizacijom ili tabuliranjem rješenja). Dinamičko programiranje razvio je u ranim 1950. godinama američki matematičar

Richard Bellman. Bellman je uveo pojam **prokletstva dimenzionalnosti**, koji smo spomenuli u kontekstu izračuna udaljenosti u visokodimenzijskom prostoru.

3 Konkretno, učinkovitost algoritma eliminacije varijabli ovisi o strukturi Bayesove mreže i o redoslijedu kojim eliminiramo varijable, kao što je pokazao naš primjer. Broj računalnih operacija ograničen je odozgo tzv. **širinom stabla** (engl. *tree width*) Bayesove mreže. Neformalno, širina stabla govori nam koliko je graf različit od stabla. Malo formalnije, širina stabla je broj čvorova u najvećoj kliki (potpuno povezanom podgrafu) grafa umanjena za jedan. Tako je stablasta širina grafa koji je stablo, a također i grafa koji je lanac, jednaka 1. Nažalost, pokazuje se da je nalaženje optimalnog redoslijeda eliminacije varijabli, koji bi dao minimalan broj računalnih operacija, NP-težak problem. U praksi stvari nisu tako crne: razvijeni su postupci za neke specifične strukture Bayesovih mreža, kao i heuristički postupci. Više u [Koller and Friedman, 2009] (poglavlje 9.4).

4 **Varijacijsko zaključivanje** (engl. *variational inference*) jedan je od osnovnih alata strojnog učenja i dio mnogih naprednih algoritmama strojnog učenja. Posebno su popularne kod generativnih dubokih modela, npr. **varijacijski autoenkoder** [Doersch, 2016]. Varijacijske metode problem zaključivanja svode na **problem optimizacije** u visokodimenzijskome prostoru – zato se te metode često opisuju sloganom "zaključivanje kao optimizacija" (engl. *inference as optimization*). Cilj takve optimizacije jest iz ograničene familije jednostavnijih distribucija pronaći onu distribuciju q koja je traktabilna (što će reći da se da izraziti u zatvorenoj formi), a opet što sličnija pravoj (ciljnoj) aposteriornoj distribuciji p. Distribucija q naziva se **prijedložna distribucija** (engl. *proposal distribution*) Sličnost između prijedložne i ciljne distribucije tipično se mjeri Kullback-Leibler divergencijom, koju smo bili spomenuli prošli put. Nakon optimizacije, probabilistički se upiti onda provode nad prijedložnom (i jednostavijom) distribucijom q, a ne nad ciljnom (i mnogo složenijom) distribucijom p. Dobar uvod u varijacijske metode možete naći u [Fox and Roberts, 2012] i [Blei et al., 2017].
Za razliku od (također popularnih) metoda uzorkovanja, o kojima ćemo pričati u nastavku, varijacijske su metode u načelu brže te računalno atraktivnije u smisu da se mogu implementirati tehnikama gradijentnog spusta i da se daju paralelizirati, što je i razlog zašto su ove tehnike danas posebno popularne u dubokom učenju. S aspekta računalne teorije učenja, varijacijske metode također imaju prednost nad metodama uzorkovanja jer su prikladnije za teorisjku analizu ograda na točnost. S druge strane, za razliku od metoda uzorkovanja, varijacijske metode rijetko nalaze globalno optimalno rješenje zbog ograničenja da prijedložna distribucija mora biti jednostavna. Zbog toga su predloženi različiti pristupi za složeniju aproksimaciju ciljne distribucije, a ti su pristupi nedavno objedinjeni u obećavajući radni okvir pod nazivom **normalizirajući tokovi** (engl. *normalizing flows*) [Papamakarios et al., 2019]. Konkretnu usporedbu varijacijskih metoda i metoda uzorkovanja (konkretno, metode MCMC; v. bilješku ispod) možete naći u [Salimans et al., 2015]. Predložene su i kombinacije varijacijskih metoda i metoda uzorkovanja; npr. vidi [Wolf et al., 2016].

5 Općenito, **Monte Carlo s Markovljevim lancem** (engl. *Markov chain Monte Carlo, MCMC*) naziv je za sve metode koje procjenjuju parametre složenih aposteriornih distribucija uzorkovanjem iz poznatih apriornih distribucija. "Monte Carlo" (prema poznatom casinu u Monacu, u kojemu je James Bond bio rado viđen gost još od "Nikad ne reci nikad") odnosi se na činjenicu da te metode koriste slučajno uzorkovanje. "Markovljev lanac" (prema ruskom matematičaru Andreyu Markovu, po kojemu je Judea Pearl uveo naziv "Markovljev omotač", a koji smo spomenuli prošli put) odnosi se na činjenicu da novi uzorak ovisi samo o prethodnom uzorku, a ne i o svim prethodnim uzorcima. O naziv MCMC više možete pročitati ovdje: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6874253/`

6 Algoritam uzorkovanja **Metropolis-Hastings** predložo je grčko-američki fizičar Nicholas Constantine Metropolis 1953. godine, radeći u Los Alamosu na poznatom projektu Mahattan, zajedno s John von Neumannom i poljsko-američkim matematičarom Stanisławom Ulamom, a 1970. godine algoritam je razradio kanadski statističar Wilfred Hastings. Usput, Metropolis je taj koji je za algoritme slučajnog uzorkovanja uveo naziv "Monte Carlo", navodno kao internu šalu na Ulamov račun. Algoritam **Gibbsovog uzorkovanja** nazvan je tako u čast američkog fizičara Josiah Willard Gibbs, jednog od osnivača statističke fizike, a predložili su ga 1984. godine (mnogo godina nakon Gibbsove smrti) američki matematičari Stuart i Donald Geman, braća koja su vrlo poznata po svojim doprinosima strojnom učenju.

7 **Skrivene varijable** već smo spomenuli kada smo govorili o probabilističkim upitima nad PGM-ovima.

Međutim, primijetite da skrivene varijable upita nisu isto kao i skrivene varijable modela. Naime, skrivene varijable upita su one varijable koje nisu opažene u trenutku kada provodimo upit. Međutim, to ne znači da te varijable nisu bile opažene onda kada smo učili model. S druge strane, skrivene varijable modela su uvijek skrivene. To su varijable koje ne možemo opažati u podatcima, jer ne postoje u podatcima, ali za koje pretpostavljamo da inače postoje, bilo u konkretnom ili apstraktnom smislu. Budući da te varijable ne opažamo, o njihovim vrijednostima **zaključujemo** na temelju opaženih varijabli, i to pomoću modela koji u odnos dovodi opažene i skrivene varijable.
Inače, modele sa skrivenim varijablama također nazivamo **model latentnih varijabli** (engl. *latent variable models, LVM*). Modeli latentnih varijabli redovito se koriste u statistici i strojnom učenju, s primjenama u područjima kao što su psihologija, medicina i ekonomija. Npr., poznati peterofaktorski model ličnosti iz psihologije (engl. *Big Five*) dobiven je modelom latentnih varijabli (konkretno, faktorskom analizom) iz jezičnih podataka.

[8] Primjer je preuzet iz [Murphy, 2012], poglavlje 11.

[9] Primijetite da kod EM-algoritma **zaključivanje** koristiom kao potproceduru algoritma učenja. Zato je bitno da zaključivanje bude učinkovito (najčešće je to onda aproksimativno zaključivanje).

[10] Detaljnije u [Murphy, 2012], poglavlje 11.3.

Literatura

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

E. V. Denardo. *Dynamic programming: models and applications*. Courier Corporation, 2012.

C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

T. Salimans, D. Kingma, and M. Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

C. Wolf, M. Karl, and P. van der Smagt. Variational inference with hamiltonian monte carlo. *arXiv preprint arXiv:1609.08203*, 2016.