

Metode Bayesovih neuronskih mreža u kompjuterskom vidu

Kanižaj, Karla

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:146396>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-13**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Karla Kanižaj

METODE BAYESOVIH NEURONSKIH
MREŽA U KOMPJUTERSKOM VIDU

Diplomski rad

Voditelj rada:
prof.dr.sc. Luka Grubišić

Zagreb, rujan, 2018.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Ovaj rad posvjećujem tati, mami i sestri. Vaša neizmijerna ljubav, vjera u mene i strpljenje su me doveli do ovog uspjeha. Hvala Vam.

Sadržaj

Sadržaj	iv
Uvod	2
1 Probabilistički grafički modeli	3
1.1 Neusmjereni graf	4
1.2 Usmjereni graf	5
2 Bayesove mreže	9
2.1 Bayesov teorem	10
2.2 Kauzalne mreže i D-separacija	12
2.3 Definicija Bayesovih mreža	19
2.4 Lančano pravilo za Bayesove mreže	23
2.5 Dodavanje znanja	25
3 Prepoznavanje i praćenje objekata	26
3.1 Problem praćenja objekata	26
3.2 Modeli kretanja	27
3.3 Procjena puta objekata	27
3.4 Bayesovo filtriranje i filteri	27
4 Prepoznavanje i praćenje lica	30
4.1 Modeliranje prostora stanja praćenog objekta	30
4.2 Problem prepoznavanja i praćenja lica	36
Bibliografija	42

Uvod

Sposobnost ljudi da uče opažanjem je primarni izvor našeg znanja o svijetu oko nas. Tako učimo klasificirati objekte i sve to samo na temelju primjera koje vidimo, a ne na temelju skupa pravila za klasifikaciju. Iskustvo nas uvelike uči o tome kako stvarati predikcije, kao na primjer predvidjeti nevrijeme na temelju izgleda neba ili pretpostaviti nepoznate kvantitativne vrijednosti kao što je procjena težine nekog predmeta i to sve samo na temelju promatranja. Bez te sposobnosti učenja iz empirijskih podataka, ne bismo bili u mogućnosti funkcionirati u svakodnevnom životu.

Postoji mnoštvo razloga za razvoj teorije ili metodologije učenja, a u području računalne znanosti se želi razumjeti kako je učenje i zaključivanje na temelju danih podataka moguće u računalnom sustavu.

Temelji teorije vjerojatnosti sežu u 16. stoljeće, kada je Gerolamo Cardano¹ započeo formalnu analizu igara na sreću, nakon čega je u 17. stoljeću uslijedio ključni razvoj zahvaljujući Pierru de Fermatu² i Blaise Pascalu³. U početku se promatrala samo diskretna vjerojatnost, a metode su bile kombinatorne. Temelji suvremene teorije vjerojatnosti, s teoretskim podlogama postavljeni su od strane Andreye Kolmogorova⁴ tridesetih godina 20. stoljeća. Nama je posebno važno 18. stoljeće kada nam je Bayesov teorem omogućio korištenje modela koji nam kaže kako iz uvjetne vjerojatnosti događaja A , uz dani događaj B , odrediti vjerojatnost događaja B , uz dani događaj A . Ovaj način zaključivanja je ključan za korištenje grafičkog modela te od tuda dolazi i naziv Bayesove mreže. [13]

Bayesove mreže, poznate kao vjerojatnosne mreže, pripadaju skupini probabilističkih grafičkih modela. Probabilistički grafički modeli su grafovi u kojima čvorovi predstavljaju slučajne varijable, a bridovi predstavljaju vjerojatnosne ovisnosti među odgovarajućim slučajnim varijablama, stoga pružaju kompaktan prikaz distribucije vjerojatnosti. Ove uvjetne zavisnosti u grafu se često procjenjuju koristeći poznate statističke i računalne metode. Dakle, Bayesove mreže kombiniraju principe iz područja teorije grafova, teorije vjerojatnosti, računalne znanosti te statistike. Bayesove mreže su matematički strogo defini-

¹ talijanski matematičar, fizičar, liječnik, filozof i astrolog, (1501. - 1576.).

² francuski matematičar i pravnik, (1601. - 1665.).

³ francuski filozof, matematičar i fizičar, (1623. - 1662.).

⁴ sovjetski matematičar, (1903. - 1987.).

rane i intuitivno razumljive. Postoje dvije osnovne vrste probabilističkih grafičkih modela, neusmjereni i usmjereni. Neusmjereni grafički modeli su također poznati kao Markovljeve mreže ili Markovljeva slučajna polja. Bayesove mreže odogovaraju usmjerenj strukturi grafički modela poznatoj kao *usmjereni aciklički graf*

(*Directed Acyclic Graph* – DAG).

Bayesove neuronske mreže (BNN) se odnose na proširenje standardnih mreža sa uvjetnim zaključivanjem. Problem treniranja standardnih neuronskih mreža je taj što koriste procjenu metodom maksimalne vjerodostojnosti te time često utječu na pretreniranost. Iz tog razloga se uvodi regularizacija kojoj je u našem slučaju ekvivalent uvođenje vjerojatnosti na težine. Uvjetno zaključivanje je vrlo izazovno iz pogleda modeliranja i računanja. Bayesove neuronske mreže koriste upravo taj pristup. U ovom radu ćemo vjerojatnosnu aparaturu prikazati na primjeru jednostavnijih Bayesovih mreža koje su samo kompaktan zapis združene tablice vjerojatnosti.

Jedan od nama važnijih problema je prepoznavanje i praćenje objekata što ćemo promotriti sa gledišta Bayesovih neuronskih mreža. Navesti ćemo Bayesove filtere, od kojih ćemo veliku pozornost u poglavlju 4.1 pridati Kalmanovom filteru koji zahtjeva da je model linearan. Kalmanov filter ćemo koristiti umjesto složene strukture Bayesove neuronske mreže. Promotriti ćemo ga u vidu rješavanja problema prepoznavanja i praćenja lica na videozapisu. To je vrlo važan problem za analizu videozapisa upravo iz razloga što su najvažniji objekti u većini videozapisa ljudi. Problem i rješenje su opisani u 4.2.

Poglavlje 1

Probabilistički grafički modeli

Probabilistički grafički modeli (PGM), prema [13] su važni u problemima učenja te su se pokazali kao odabrana metoda za modeliranje vjerojatnosti u mnogim područjima kao što su kompjuterski vid, obrada govora, vremenski nizovi, sekvencijalno modeliranje podataka, kognitivne znanosti, bioinformatika, probabilistička robotika, obrada signala te u području umjetne inteligencije općenito. Jedan od razloga za ovako široku primjenu je to što ovakvo modeliranje omogućuje prijenos koncepata i ideja među različitim područjima primjene. Dugi razlog izražen u [23] je taj da je razumijevanje razumnog zaključivanja prirodno ugrađeno u sintaksu izračunavanja vjerojatnosti.

PGM kombiniraju teoriju vjerojatnosti i teoriju grafova te daju kompaktan grafički prikaz zajedničke distribucije vjerojatnosti uključujući uvjetne zavisnosti među slučajnim varijablama. Pretpostavke vezane uz uvjetne zavisnosti olakšavaju računalni dio.

Grafički modeli pružaju tehnike za rješavanje dvaju problema u primijenjenoj matematici i inženjerstvu, a to su nesigurnost i složenost [10].

Mnogi vrlo poznati statistički modeli, kao što su Bayesove mreže, skriveni Markovljevi modeli, Kalmanovi filtri, Boltzmannov stroj te mnoštvo drugih, mogu se, između ostalog, prikazati grafičkim modelima.

Mnogi algoritmi uključuju vrlo velike raspodjele vjerojatnosti preko velikog broja slučajnih varijabli. Često, ove distribucije vjerojatnosti uključuju izravne interakcije između relativno malo slučajnih varijabli. Korištenje jedne funkcije za opisati cijelu zajedničku distribuciju vjerojatnosti može biti vrlo neučinkovito, gledajući računalno i statistički. Stoga, umjesto korištenja jedne funkcije za predstavljanje raspodjele vjerojatnosti, možemo podijeliti raspodjelu vjerojatnosti u mnoge faktore koje ćemo izmnožiti.

Na primjer, pretpostavimo da imamo tri slučajne varijable: A , B i C . Pretpostavimo da A utječe na vrijednost B i B utječe na vrijednost C , ali da su A i C neovisni o danom B . Možemo prikazati raspodjelu vjerojatnosti nad sve tri varijable kao produkt distribucije

vjerojatnosti preko dvije varijable:

$$P(A, B, C) = P(A)P(B|A)P(C|B)$$

Ove faktorizacije mogu znatno smanjiti broj potrebnih parametara za opisivanje distribucije. Svaki faktor koristi broj parametara koji je eksponencijalan u broju varijabli u faktoru. To znači da možemo znatno smanjiti troškove prikazivanja distribucije ako usprijemo pronaći faktorizaciju u distribucije na manji broj varijabli. Upravo takve faktorizacije možemo opisati pomoću grafova.

Definicija 1.0.1. *Graf \mathcal{G} je uređeni par $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$. $V = V(\mathcal{G})$ je neprazan konačan skup, čiji su elementi **vrhovi (čvorovi)** od \mathcal{G} . $E = E(\mathcal{G})$ je konačan skup neuređenih parova iz skupa V (ne nužno različitih) koje nazivamo **bridovima**.*

Kada grafom predstavljamo faktorizaciju raspodjele vjerojatnosti, takav graf nazivamo **probabilistički grafički model** ili samo **grafički model**.

Grafički modeli koriste graf \mathcal{G} u kojem svaki čvor grafa predstavlja slučajnu varijablu, a brid koji povezuje dvije slučajne varijable znači da se distribucija vjerojatnosti može prikazati direktnim interakcijama između te dvije slučajne varijable. Nadalje pretpostavljamo da su čvorovi grafa \mathcal{G} , $V = \{X_1, \dots, X_N\}$. Bilo koja dva brida $X_i, X_j, i \neq j$ koji odgovaraju vezama između slučajnih varijabli, mogu biti *usmjereni* ili *neusmjereni*. Neusmjereni brid između čvora i i čvora j se označava kao $(X_i - X_j)$, a usmjereni brid od čvora i do čvora j se označava kao $(X_i \rightarrow X_j)$. Neusmjereni brid je simetrična relacija između čvorova, odnosno, brid $(X_i - X_j)$ je jednak bridu $(X_j - X_i)$, a usmjereni brid predstavlja asimetričnu relaciju. Smatramo da ne postoje čvorovi između kojih postoje obje vrste bridova te ne postoji brid koji povezuje neki čvor sa samim sobom. Dakle, osnovna razlika između usmjerenih i neusmjerenih bridova je ta što je usmjereni brid uređeni par čvorova.

Obzirom na vrstu brida, razlikujemo za nas dva bitna tipa grafova, usmjereni i neusmjereni.

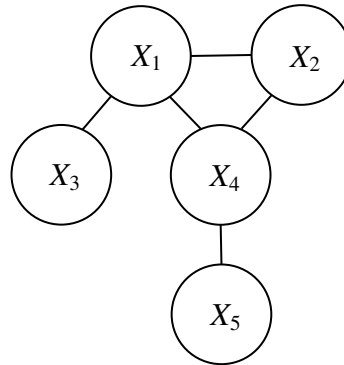
1.1 Neusmjereni graf

Definicija 1.1.1. (*Neusmjereni graf*) *Za graf $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ kažemo da je neusmjereni ako je svaki brid $e \in E$ neusmjeren.*

Neusmjereni grafički modeli obično se nazivaju Markovljeva slučajna polja ili Markovljeve mreže. Te mreže pružaju jednostavnu definiciju neovisnosti između dva različita čvora grafa temeljenu na konceptu Markovljevog pokrivača. Markovljeve mreže su popularne u područjima kao što su statistička fizika i kompjuterski vid. [10] [29]

Primjer 1.1.2. Pogledajmo sliku 1.1. Za prikazani neusmjereni graf \mathcal{G} te njegove čvorove $V = \{X_1, \dots, X_5\}$, pripadni bridovi su:

$$E = \{X_1 - X_2\}, \{X_1 - X_3\}, \{X_1 - X_4\}, \{X_2 - X_4\}, \{X_4 - X_5\}$$



Slika 1.1: Neusmjereni graf

1.2 Usmjereni graf

Definicija 1.2.1. (*Usmjereni graf*) Za graf $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ kažemo da je usmjereni ako je svaki brid $e \in E$ usmjeren.

Usmjereni grafički modeli, poznatiji su kao Bayesove mreže, vjerojatnosne mreže, generativni modeli. Ako graf sadrži brid $\{X_1 \rightarrow X_2\}$, intuitivno možemo shvatiti da X_1 uzrokuje X_2 . Kad govorimo o vezama unutar usmjerenog grafa, ako postoji brid od čvora X_i do čvora X_j , kažemo da je X_j *dijete* čvora X_i , a X_i je *roditelj* od X_j .

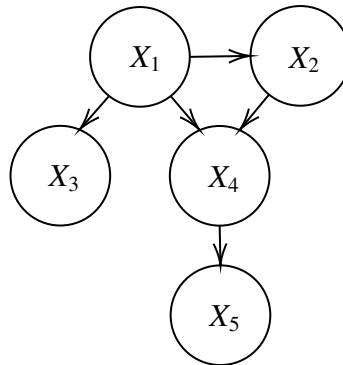
Primjer 1.2.2. Pogledajmo sliku 1.2. Za prikazani usmjereni graf \mathcal{G} te njegove čvorove $V = \{X_1, \dots, X_5\}$, pripadni bridovi su:

$$E = \{X_1 \rightarrow X_2\}, \{X_1 \rightarrow X_3\}, \{X_1 \rightarrow X_4\}, \{X_2 \rightarrow X_4\}, \{X_4 \rightarrow X_5\}$$

Usmjereni modeli se prikazuju faktorizacijom u distribucije uvjetne vjerojatnosti, točnije, usmjereni model sadrži jedan faktor za svaku slučajnu varijablu X_i u distribuciji i taj faktor sadrži uvjetnu distribuciju na X_i , s obzirom na roditelje od X_i .

Primjer 1.2.3. Usmjereni graf na slici 1.2, sa slučajnim varijablama $\{X_1, X_2, X_3, X_4, X_5\}$, odgovara vjerojatnosnoj distribuciji koja može biti faktorizirana na sljedeći način:

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1, X_2)P(X_5|X_4)$$



Slika 1.2: Usmjereni graf

Bridovi predstavljaju direktnu vezu između dva čvora, dok *put* i *staza* opisuju indirektnu vezu između više čvorova.

Definicija 1.2.4. (Put) Neka je $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ graf. Niz čvorova $Q = (X_1, \dots, X_n)$, $X_1, \dots, X_n \in V$ je put od čvora X_1 do X_n , ako

$$\forall i \in \{1, \dots, n-1\} : (X_i \rightarrow X_{i+1}) \in E \vee (X_i - X_{i+1}) \in E.$$

Definicija 1.2.5. (Usmjereni put) Neka je $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ graf. Za niz čvorova $Q = (X_1, \dots, X_n)$, $X_1, \dots, X_n \in V$ kažemo da je usmjereni put od čvora X_1 do X_n , ako

$$\exists i \in \{X_1, \dots, X_n\} : (X_i \rightarrow X_{i+1}) \in E,$$

odnosno ako postoji barem jedan usmjereni brid na putu.

Primjer 1.2.6. Promotrimo ponovo sliku 1.2.

- (a) Niz $\{X_1, X_2, X_4, X_5\}$ je put od $\{X_1\}$ do $\{X_5\}$.
- (b) Niz $\{X_1, X_4, X_5\}$ je usmjereni put od $\{X_1\}$ do $\{X_5\}$, ali ne postoji put od čvora $\{X_3\}$ do $\{X_5\}$ zbog usmjerenih bridova.

Poseban tip putova su usmjereni ciklusi.

Definicija 1.2.7. (Usmjereni ciklus) Neka je $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ graf, $X_i \in V$ te P usmjereni put od X_i do X_i . Tada P nazivamo usmjereni ciklus.

Koristeći ovu definiciju možemo definirati važan tip usmjerenih grafova:

Definicija 1.2.8. (Usmjereni aciklički graf) Graf $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ je usmjereni aciklički graf (DAG) ako ne sadrži usmjerene cikluse.

Primjer 1.2.9. *Primjer jednog usmjerenog acikličkog grafa je dan na slici 1.2.*

Svaki usmjereni aciklički graf ima barem jedan topološki uređaj. Sada neformalno uvodimo svojstvo nezavisnosti "budućnosti" od "prošlosti", uz fiksiranu "sadašnjost" koje nazivamo **Uređajno Markovljevo svojstvo (UMS)**: svaki čvor X_k ovisi samo o roditeljima:

$$X_k \perp \text{pred}(X_k) \setminus \text{pa}(X_k) \mid \text{pa}(X_k),$$

gdje je $\text{pred}(X_k)$ skup prethodnika čvora X_k po topološkom uređaju, a $\text{pa}(X_k)$ označava čvorove roditelja čvora X_k . Ukratko, ako znamo "sadašnjost", tada "prošlost" nema utjecaja na "budućnost".

Primjer 1.2.10. *Pogledajmo korištenje faktorizacije i UMS-a na primjeru sa slike 1.3.*

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2, X_3)$$

Uvjetne nezavisnosti koje proizlaze iz UMS-a:

$$X_k \perp \text{pred}(X_k) \setminus \text{pa}(X_k) \mid \text{pa}(X_k)$$

$$X_2 \perp \{X_1\} \setminus \{X_1\} \mid \{X_1\}$$

$$X_3 \perp \{X_1, X_2\} \setminus \{X_1\} \mid \{X_1\} \Rightarrow X_2 \perp X_3 \mid X_1$$

(1.1)

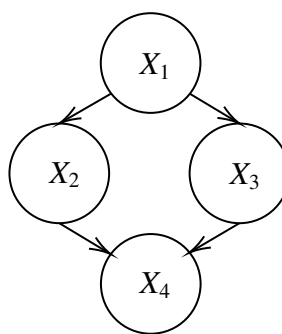
$$X_4 \perp \{X_1, X_2, X_3\} \setminus \{X_2, X_3\} \mid \{X_1, X_2\} \Rightarrow X_1 \perp X_4 \mid X_2, X_3$$

(1.2)

Faktorizacijom i korištenjem uvjetne nezavisnosti slijedi:

$$\begin{aligned} P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2, X_3) &= P(X_1, X_2)P(X_3|X_1)P(X_4|X_2, X_3) \\ &\stackrel{(1.1)}{\Rightarrow} P(X_1, X_2)P(X_3|X_1, X_2)P(X_4|X_2, X_3) \\ &= P(X_1, X_2, X_3)P(X_4|X_2, X_3) \\ &\stackrel{(1.2)}{\Rightarrow} P(X_1, X_2, X_3)P(X_4|X_1, X_2, X_3) \\ &= P(X_1, X_2, X_3, X_4) \end{aligned}$$

Bayesove mreže odgovaraju upravo strukturi usmjerenog acikličkog grafa. Struktura acikličkog grafa osigurava da ne postoji čvor koji može biti svoj vlastiti predak ili vlastiti potomak. To je nužno za faktorizaciju združene vjerojatnosti čvorova. Iako strelice predstavljaju izravnu kauzalnu povezanost između varijabli, proces zaključivanja Bayesovim mrežama može teći širenjem informacija u bilo kojem smjeru.[24]



Slika 1.3: Usmjereni aciklički graf

Poglavlje 2

Bayesove mreže

Bayesove mreže (BN) postale su vrlo popularan model te se koriste za različite primjene u mnogim područjima kao što su strojno učenje, obrada prirodnog jezika, prepoznavanje govora, obrada signala, bioinformatika, medicinska dijagnoza. Iako samo ime govori, Bayesove mreže ne uključuju nužno upotrebu Bayesove statistike. Ponekad se koriste i druge metode za procjenu parametara, a čvorovi mogu predstavljati ne samo slučajne varijable nego i hipoteze i uvjerenja. Njihova struktura je intuitivno privlačna i prikladna za prikazivanje kauzalne i vjerojatnostne semantike. Kao što je naznačeno u [7], ova struktura je idealna za kombiniranje prethodnog znanja, koje često dolazi u kauzalnom obliku te promatranje podataka. Bayesove mreže se mogu koristiti, čak i u slučaju nedostatka dijela podataka, za učenje kauzalnih veza i stjecanje razumijevanja raznih domena problema i predviđanje budućih događaja.

Bayesove mreže imaju dugu povijest u području statistike i počeci korištenja se mogu naći i u ovom radu [16]. U prvoj polovici 1980-ih uvodi se njihova primjena u područje stručnih sustava kroz sljedeće radove: [22], [28]. Neki od prvih primjena Bayesovih mreža u stvarnoj produkciji bile su *Munin* [1] i *Pathfinder* [8].

Bayesova mreža predstavlja kauzalnu vjerojatnosnu relaciju među skupom slučajnih varijabli, njihovom uvjetnom ovisnosti i pruža kompaktan prikaz zajedničke distribucije vjerojatnosti. [18] Sastoji se od dva glavna dijela: usmjerenog acikličkog grafa i skupa distribucija uvjetnih vjerojatnosti.

U ovom poglavlju ponoviti ćemo važne definicije iz područja vjerojatnosti te uvesti Bayesov teorem, potom uvesti i kauzalne mreže, a zatim definirati Bayesove mreže kao kauzalne mreže s jačinom kauzalnih bridova reprezentiranih kao uvjetne vjerojatnosti. Na posljetku ćemo prikazati lančano pravilo koje čini Bayesove mreže snažnim alatom za reprezentiranje domena s nasljednom nezavisnošću.

2.1 Bayesov teorem

Engleski statističar i filozof Thomas Bayes (1702.–1762.) dokazao je čuveni teorem teorije vjerojatnosti koji po njemu nazivamo Bayesov teorem ili češće Bayesova formula. Cijeli koncept Bayesovih mreža temelji se na Bayesovom teoremu koji pomaže izraziti vjerojatnost hipoteze ako znamo da se dogodio neki događaj. Odnosno, daje nam mogućnost korištenja modela koji nam daje vjerojatnost događaja A uz uvjet B u svrhu izračunavanja vjerojatnosti događaja B uz uvjet A . Takva vrsta zaključivanja je ključna za korištenje grafičkih modela te objašnjava izbor imena Bayesove mreže.

Prije samog Bayesovog teorema, definirat ćemo uvjetnu vjerojatnost i navesti formulu potpune vjerojatnosti.

Definicija 2.1.1. *Neka je (Ω, \mathcal{F}, P) proizvoljan vjerojatnosni prostor i $A \in \mathcal{F}$ takav da je $P(A) > 0$. Definirajmo funkciju $P_A : \mathcal{F} \rightarrow \{0, 1\}$ ovako:*

$$P_A(B) = P(B|A) = \frac{P(A \cap B)}{P(A)}, \quad B \in \mathcal{F}. \quad (2.1)$$

P_A je vjerojatnost na \mathcal{F} i nju zovemo **uvjetna vjerojatnost uz uvjet A** . Broj $P(B|A)$ zovemo **vjerojatnost od B uz uvjet da se A dogodio** ili kraće, **vjerojatnost od B uz uvjet A** . Prema tome, svakom događaju A za koji je $P(A) > 0$ pridružuje se preko relacije (2.1) vjerojatnosni prostor $(\Omega, \mathcal{F}, P_A)$.

Napomena 2.1.2. *Analogno se definira i $P(A|B)$ pa vrijedi pravilo umnoška vjerojatnosti:*

$$P(A \cap B) = P(B) P(A|B) = P(A) P(B|A), \quad A, B \in \mathcal{F}. \quad (2.2)$$

Za događaje A i B , za koje je $P(A) > 0, P(B) > 0$, može se dogoditi da informacija o događaju B ne mijenja znanje o događaju A . To znači da $P(A|B) = P(A)$, odnosno $P(B|A) = P(B)$. Svaka od tih jednakosti ekvivalentna je s $P(A \cap B) = P(A)P(B)$. U tom slučaju ćemo reći da su događaji A i B nezavisni.

Definicija 2.1.3. *Neka je (Ω, \mathcal{F}, P) proizvoljan vjerojatnosni prostor:*

(a) *Događaji $A, B \in \mathcal{F}$ su nezavisni ako vrijedi*

$$P(A \cap B) = P(A)P(B).$$

(b) *Događaji $A, B \in \mathcal{F}$ su uvjetno nezavisni uz dani $C \in \mathcal{F}, P(C) > 0$ i pišemo $A \perp B|C$, ako vrijedi*

$$P(A \cap B|C) = P(A|C)P(B|C).$$

Definicija 2.1.4. *Konačna ili prebrojiva familija $H_i, (i = 1, 2, \dots)$, događaja u vjerojatnosnom prostoru (Ω, \mathcal{F}, P) jest **potpun sistem događaja** ako je $H_i \neq \emptyset$ za svako i , $H_i \cap H_j = \emptyset$ za $i \neq j$ (tj. događaji H_i uzajamno se isključuju) i $\bigcup_i H_i = \Omega$.*

Drugim riječima, potpun sistem događaja konačna je ili prebrojiva particija skupa Ω s tim da su elementi particije događaji.

Teorem 2.1.5. (formula potpune vjerojatnosti) *Neka je $H_i, (i = 1, 2, \dots)$, potpun sistem događaja u vjerojatnosnom prostoru (Ω, \mathcal{F}, P) . Tada za proizvoljno $A \in \mathcal{F}$ vrijedi:*

$$P(A) = \sum_i P(H_i)P(A|H_i). \quad (2.3)$$

Dokaz. Za $A \in \mathcal{F}$ vrijedi

$$\begin{aligned} P(A) &= P(A \cap \Omega) = P\left(A \cap \left(\bigcup_i H_i\right)\right) = P\left(\bigcup_i (A \cap H_i)\right) \\ &= \sum_i P(A \cap H_i) = \sum_i P(H_i)P(A|H_i). \end{aligned}$$

□

Na formulu potpune vjerojatnosti nadovezuje se Bayesova formula. Pretpostavimo da imamo zadan potpun sustav događaja i poznate vjerojatnosti svake od hipoteza $P(H_i) > 0, i = 1, 2, \dots$, te znamo da se nakon izvođenja pokusa pojavio događaj A kao njegov ishod. Želimo svakoj od hipoteza $H_i, i = 1, 2, \dots$, pridružiti vjerojatnosti $P(H_i|A)$. Iz (2.2) slijedi

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}.$$

Uzmimo sada da je događaj B jedna od hipoteza $H_i, i = 1, 2, \dots$, na koje je skup Ω podijeljen, odnosno

$$P(H_i|A) = \frac{P(H_i)P(A|H_i)}{P(A)}.$$

Pri tome vjerojatnost događaja A najčešće računamo pomoću formule potpune vjerojatnosti (2.3). Tako smo došli do Bayesove formule.

Teorem 2.1.6. (Bayesova formula) *Neka je $H_i, (i = 1, 2, \dots)$, potpun sistem događaja u vjerojatnosnom prostoru (Ω, \mathcal{F}, P) i $A \in \mathcal{F}$ takav da je $P(A) > 0$. Tada za svako i vrijedi*

$$P(H_i|A) = \frac{P(H_i)P(A|H_i)}{\sum_j P(H_j)P(A|H_j)}. \quad (2.4)$$

Dokaz. Iz (2.1), (2.2) i teorema 2.1.5 slijedi

$$P(H_i|A) = \frac{P(H_i \cap A)}{P(A)} = \frac{P(H_i)P(A|H_i)}{\sum_j P(H_j)P(A|H_j)}.$$

□

Bayesova formula daje vjerojatnost hipoteze ako znamo da se dogodio neki događaj.

2.2 Kauzalne mreže i D-separacija

Kauzalna mreža se sastoji od skupa slučajnih varijabli i skupa usmjerenih bridova među varijablama. Ta struktura je upravo usmjereni graf. Varijable mogu imati proizvoljan broj stanja, ali u obzir uzimamo samo one s konačnim brojem stanja.

U kauzalnoj mreži, varijabla predstavlja skup mogućih stanja odnosa te je varijabla u točno jednom od svojih stanja i taj podatak nam je možda nepoznat.

Kauzalne mreže se mogu koristiti za praćenje kako promjena vjerojatnosti jedne varijable može utjecati na promjenu vjerojatnosti drugih varijabla. Ovdje ćemo pokazati skup pravila za takav način zaključivanja. Sada uvodimo tri jednostavna grafa koja ilustriraju tri različite vrste veza i time implicirane uvjetne neovisnosti. To su redom ulančana, divergirajuća i konvergirajuća veza.

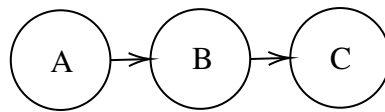
Ulančana veza

Mreža koja prikazuje *ulančanu vezu* prikazana je na slici 2.1. Ovdje A utječe na B , a B potom utječe na C . Očito, znanje o A će utjecati na sigurnost od B , koja dalje utječe na sigurnost od C . Slično, znanje o C će utjecati na sigurnost od A preko B . S druge strane, ako je poznato stanje od B , tada je veza blokirana te A i C postaju neovisne. Kažemo da su A i C postale *D-separirane* za dani B .

Kad je stanje varijable poznato, kažemo da je varijabla *opažena*.

- Ako varijabla B nije opažena, tada su A i C ovisni.
Informacija se može prenositi između A i C ukoliko B nije opažena.
- Ako je varijabla B opažena, tada su A i C neovisni.
Informacija se ne može prenositi između A i C ukoliko je B opažena. Opaženost od B blokira informacijski put.

Dakle, ako je varijabla B **opažena**, onda su čvorovi odvojeni, inače su povezani. Informacija se može prenositi ulančanom vezom ukoliko varijabla u vezi nije opažena.



Slika 2.1: Ulančana veza. Ukoliko je B opažena, blokira komunikaciju među A i C .

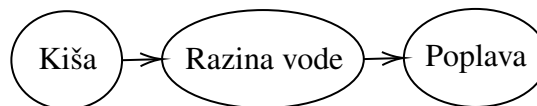
Primjer 2.2.1. Za dani primjer na slici 2.1 slijedi

$$P(A, B, C) = P(A)P(B|A)P(C|B),$$

te je uređajno Markovljevo svojstvo (UMS):

$$C \perp A | B \Leftrightarrow A \perp C | B$$

Primjer 2.2.2. Slika 2.2 pokazuje kauzalni model relacije između Kiša (ne, lagana, srednja, jaka), Razina vode (niska, srednja, visoka) i Poplava (da, ne). Ako ne znamo razinu vode, tada će znanje o tome da postoji poplava uvećati vjerovanje da je razina vode visoka što će povratno dati informaciju o padanju kiše. Jednak način zaključivanja je i u drugom smjeru. S druge strane, ako znamo razinu vode, tada znanje o postojanju poplave ne govori ništa o padanju kiše.

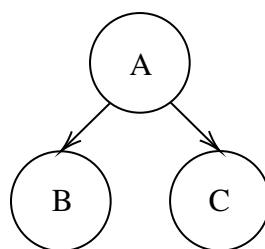


Slika 2.2: Kauzalni model za Kišu, Razinu vode i Poplavu.

Divergirajuća veza

Mreža na slici 2.3 prikazuje *divergirajuću vezu*. Utjecaj prolazi kroz svu djecu od A ukoliko nije poznato stanje od A . U suprotnom su B i C D-separirane za dani A .

- Ako varijabla A nije opažena, tada su B i C ovisni.
Informacija se može prenositi između djece od A ukoliko A nije opažena.
- Ako je varijabla A opažena, tada su B i C neovisni.
Informacija se ne može prenositi između djece od A ukoliko je A opažena. Opaženost od A blokira informacijski put.



Slika 2.3: Divergirajuća veza. Ukoliko je A opažena, blokira komunikaciju među djecom.

Dakle, ako je varijabla A **opažena**, onda su čvorovi odvojeni, inače su povezani. Informacija se može prenositi divergirajućom vezom ukoliko varijabla u vezi nije opažena.

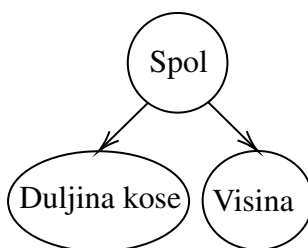
Primjer 2.2.3. Za dani primjer na slici 2.3 slijedi

$$P(A, B, C) = P(A)P(B|A)P(C|A)$$

te je uređajno Markovljevo svojstvo (UMS):

$$C \perp B | A \Leftrightarrow B \perp C | A$$

Primjer 2.2.4. Slika 2.4 pokazuje kauzalni model relacije između Spol (muško, žensko), Duljina kose (kratka, duga) i Visina ($< 168\text{cm}$, $\geq 168\text{cm}$). Ako ne znamo spol osobe, znanje o duljini kose će nam reći nešto više o spolu što će usmjeriti naše vjerovanje o visini. S druge strane, ako znamo da se radi o muškarcu, tada znanje o duljini kose ne govori ništa o visini te osobe.



Slika 2.4: Kauzalni model koji prikazuje da Spol ima utjecaj na Duljinu kose i Visinu.

Konvergirajuća veza

Mreža na slici 2.5 prikazuje *konvergirajuću vezu*. Ako ništa ne znamo o A , osim onog što može biti izvedeno iz znanja roditelja B i C , onda su roditelji neovisni: znanje o jednom

od njih ne može utjecati na sigurnost ostalih preko A . Znanje jednog mogućeg uzroka događaja ne govori nam ništa o drugim mogućim uzrocima. Međutim, ako se barem nešto zna o posljedicama, tada informacija o jednom mogućem uzroku nam može reći nešto o drugom. Ukoliko A nije opažena, varijable B i C su neovisne, ali postanu ovisne opažanjem varijable A ili bilo kojeg sljedbenika od A . Ovo je poznato kao *Berksonov paradoks ili Efekt objašnjavanja* (eng. *explaining away*): opažanje A i B smanjuje vjerojatnost za C , odnosno,

$$P(B|A) \neq P(B|A, C) \Leftrightarrow B \perp\!\!\!\perp C|A.$$

Kod konvergirajuće veze, varijable B i C se "natječu" za objašnjavanje (uzorkovanje) varijable A .

Na slici 2.6 su dva primjera konvergirajuće veze. Uočimo da u drugom primjeru varijablu A proučavamo indirektno preko informacije o F , znanje o stanju od F nam govori nešto o stanju od E koje nam potom govori nešto o stanju od A .

- Ako varijabla A ni jedan njen nasljednik nije opažen, tada su B i C neovisni. Informacija se ne može prenositi preko A kroz roditelje od A . Putuje kroz A i njene nasljednike.
- Ako je varijabla A ili neki od njenih nasljednika opažen, tada su B i C ovisni. Informacija se može prenositi preko A kroz roditelje od A ukoliko je A ili neki njen nasljednik opažen. Opaženost od A ili njenih nasljednika otvara informacijski put.

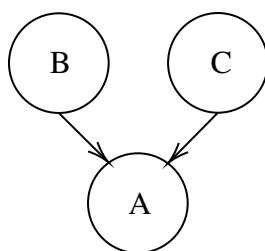
Dakle, ako je varijabla A **neopažena**, onda su čvorovi odvojeni, inače su povezani. Informacija se može prenositi konvergirajućom vezom ukoliko je varijabla u vezi ili neki njen sljedbenik opažen.

Dokaz o varijabli je tvrdnja o sigurnosti njenih stanja. Razlikujemo *čvrsti* (eng. *hard*) i *meki* (eng. *soft*) dokaz.

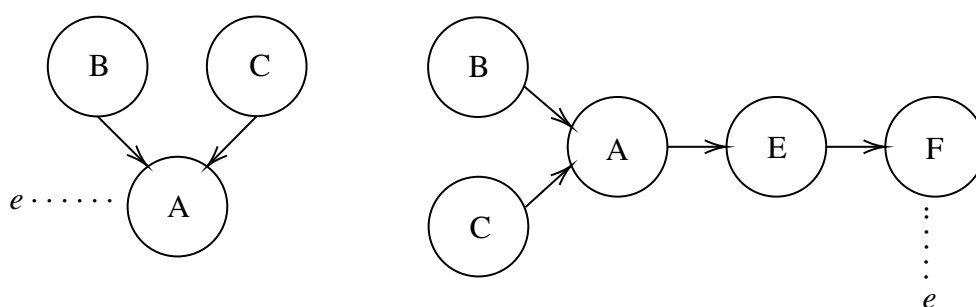
- **Čvrsti dokaz** o varijabli: Vrijednost varijable je direktno opažena.
- **Meki dokaz** o varijabli: Vrijednost varijable nije direktno opažena, ali je vrijednost njenog sljedbenika opažena.

Na primjeru 2.6, možemo reći da čvrsti dokaz o varijabli F daje meki dokaz o varijabli A . Pravila za prenošenje informacija:

- Čvrsti dokaz blokira prenošenje informacija u slučaju ulančane i divergirajuće veze.
- I meki i čvrsti dokaz su dovoljni za prenošenje informacija u slučaju konvergirajuće veze.



Slika 2.5: Konvergirajuća veza. Ukoliko A mijenja sigurnost, otvara komunikaciju među roditeljima.



Slika 2.6: Primjeri konvergirajuće veze u kojima su roditelji od A ovisni. Iscrtana linija označava uvođenje dokaza.

Primjer 2.2.5. Za dani primjer na slici 2.5 slijedi

$$P(A, B, C) = P(B)P(C)P(A|B, C)$$

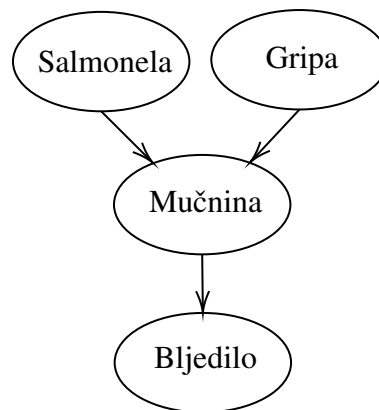
te je uređajno Markovljevo svojstvo(UMS):

$$C \perp B | \emptyset$$

Primjer 2.2.6. Slika 2.7 prikazuje kauzalne relacije između infekcije Salmonelom, Gripa, Mučnine i Bljedila. Ako ne znamo ništa o mučnini i bljedilu, tada nam informacija o tome da li je osoba zaražena Salmonelom neće reći ništa o gripu. Međutim, ako uočimo bljedilo, tada informacija da osoba nije zaražena Salmonelom će usmjeriti naše vjerovanje na to da ta osoba ima gripu.

D-separacija

Tri prethodna slučaja ulančane, divergirajuće i konvergirajuće veze pokrivaju sve načine na koje se informacije mogu prenijeti kroz varijablu. Dana pravila nam omogućuju da za bilo



Slika 2.7: Infekcija Salmonelom i gripa mogu uzrokovati mučninu, što uzrokuje bljedilo.

koji par varijabla u kauzalnoj mreži odlučimo jesu li neovisne s obzirom na informaciju. Pravila su formulirana u sljedeću definiciju.

Definicija 2.2.7. (D-separacija) Dvije različite varijable A i B u kauzalnoj mreži su D -separirane ako za sve putove između A i B , postoji posredna varijabla V , različita od A i B takva da vrijedi jedno:

- veza je ulančana ili divergirajuća te je V opažena

ili

- veza je konvergirajuća te ni V ni bilo koji nasljednik od V nije opažen

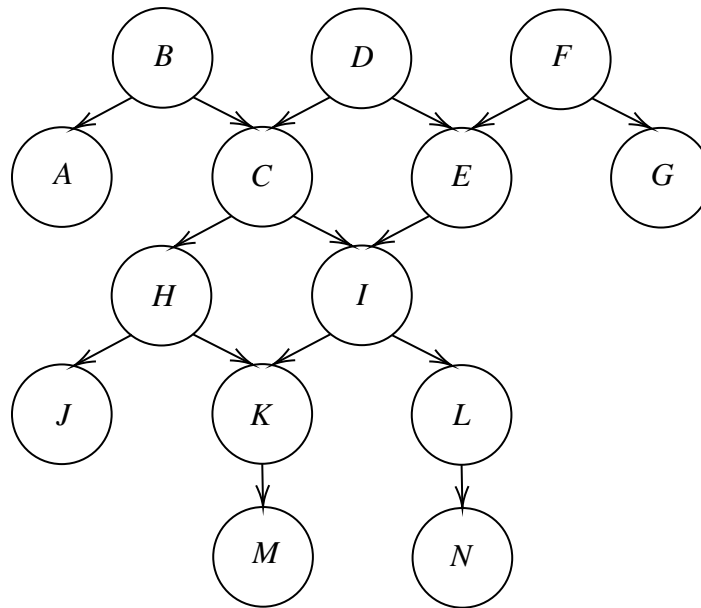
Ako A i B nisu D -separirane, nazivamo ih D -povezanima.

Ako su A i B D -separirane, tada promijene u sigurnosti od A ne utječu na sigurnost od B .

Poseban skup čvorova za čvor A nazivamo Markovljev pokrivač za A .

Definicija 2.2.8. Markovljev pokrivač varijable A je skup koji se sastoji od roditelja od A , djece od A i varijabla koje dijele dijete s A .

Markovljev pokrivač ima svojstvo da kada je opažena, varijabla A je D -separirana od ostatka mreže, to možemo vidjeti na slici 2.8.



Slika 2.8: Markovljevi pokrivači za I je $\{C, E, H, K, L\}$. Uočimo da ako su samo susjedi od I opaženi, tada J nije D-separirana od I .

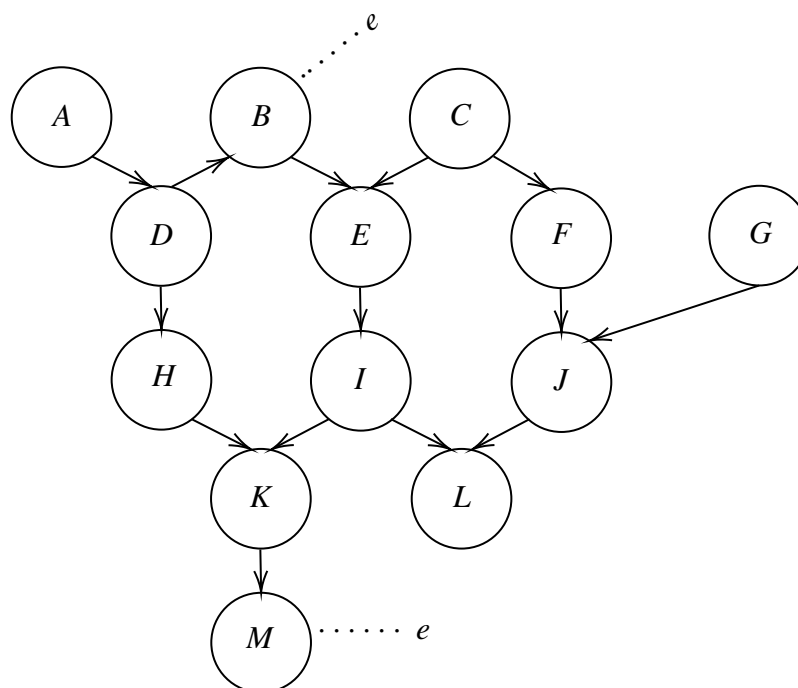
Određivanje D-separabilnosti

Iz definicije D-separabilnosti vidimo da ukoliko želimo odrediti jesu li dvije varijable A i B D-separirane za dan čvrsti dokaz na skupu varijabli C , morali bi provjeriti da li su svi putovi koji povezuju A i B D-separirani. Lakši način za provedbu ovog testa, bez uzimanja u obzir različite vrste veza je sljedeći:

- Konstruiramo graf prethodnika koji sadrži A , B te C , zajedno sa svim čvorovima od kojih postoji usmjereni put do A , B ili C . Skup *prethodnika* skupa X sastoji se od svih čvorova u X i svih prethodnika svakog od čvora u X .
- Dodajemo neusmjereni brid između svakog para čvorova sa zajedničkim djetetom te potom svaki brid zamjenimo neusmjerenim.

Rezultat ovog postupka je graf koji nazivamo *moralni graf* (eng. *moral graph*). Moralni graf nam sada može poslužiti kako bismo provjerili jesu li varijable A i B D-separirane za dani C i to na sljedeći način: ako svi putovi koji povezuju A i B presijecaju C , tada su A i B D-separirani za dani C .

Primjer 2.2.9. *Provjerimo D-separabilnosti varijable A i F uz dane dokaze o B i M na slici 2.9. Prvo konstruiramo graf prethodnika za $\{A, B, M, F\}$ što vidimo na lijevoj slici*

Slika 2.9: Kauzalna mreža s opaženim M i B .

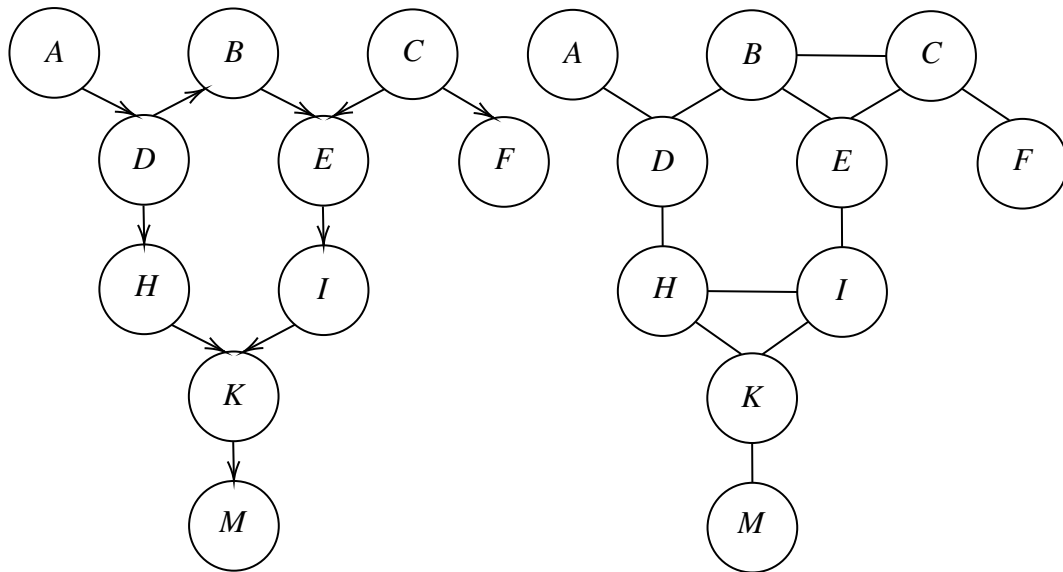
2.10. Zatim, dodajemo neusmjereni brid između svakog para čvora te svaki usmjereni brid zamijenimo neusmjerenim bridom. Na rezultirajućem grafu na desnoj slici 2.10 imamo da put $A - D - H - K - I - E - C - F$ ne siječe B i M . Stoga, A i F su D -povezani uz dane dokaze o B i M .

2.3 Definicija Bayesovih mreža

Kauzalne relacije imaju kvantitativnu stranu, posebno svoju jačinu. To možemo opisati pridavanjem brojeva bridovima. Neka je čvor A roditelj čvora B , tada bi prirodno bilo da vrijedi $P(B|A)$ jačina brida između ta dva čvora. Međutim, ako je C također roditelj čvora B , tada vrijedi i $P(B|C)$, ali te dvije uvjetne vjerojatnosti ne daju nikakvu informaciju o vezi između A i C . Potrebno nam je informacija o $P(B|A, C)$.

Definicija 2.3.1. *Bayesova mreža se sastoji od sljedećeg:*

- Skup slučajnih varijabli i skup usmjerenih bridova među varijablama.
- Svaka varijabla ima konačni skup međusobno isključivih stanja.



Slika 2.10: Lijeva slika prikazuje graf prethodnika za $\{A, B, M, F\}$, a desna slika prikazuje graf dobiven dodavanjem neusmjerenih i zamjene usmjerenih neusmjerenim bridovima.

- Varijable zajedno s usmjerenim bridovima tvore usmjereni aciklički graf.
- Svakoj varijabli A s roditeljima B_1, \dots, B_n je pridružena tablica uvjetnih vjerojatnosti $P(A|B_1, \dots, B_n)$.

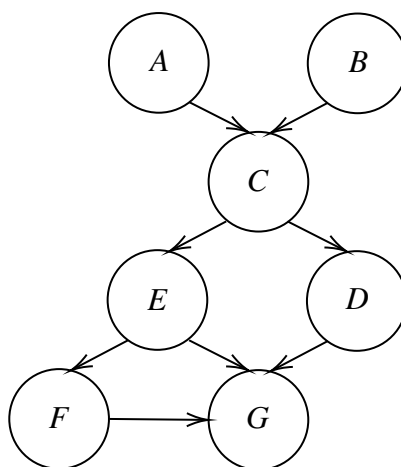
Uočimo da ako čvor A nema roditelja, tada se tablica reducira na tablicu bezuvjetne vjerojatnosti $P(A)$. Za usmjereni aciklički graf na 2.11 vjerojatnosti $P(A)$ i $P(B)$ moraju biti navedene. Pretpostavlja se da su priorne vjerojatnosti neželjeni uvod pristranosti u model te se izbjegavaju. Međutim, te vjerojatnosti su sastavni dio ljudskog razmišljanja o sigurnosti.

Definicija Bayesovih mreža ne uključuje kauzalnost i ne postoji zahtjev da bridovi predstavljaju kauzalni utjecaj. Kada gradimo strukturu modela Bayesove mreže, ne inzistiramo da bridovi idu kauzalno, ali trebamo provjeriti D-separacijska svojstva modela. Ta svojstva moraju odgovarati našoj percepciji svojstva uvjetne nezavisnosti koja su moguća u stvarnom svijetu.

To također znači da ako su A i B D-separirani za dani dokaz e , onda $P(A|e) = P(A|B, e)$.

Uvjet Bayesove mreže jest da za svaku slučajnu varijablu A vrijedi da je ona nezavisna od svake druge varijable B koja joj nije potomak, uvjetno na svoje roditelje. Ovaj uvjet smo napomenuli kao *uredajno Markovljevo svojstvo*.

Primjer 2.3.2. Proučimo primjer "Paljenje auta" dan u [21]: Jutro je i moj auto ne želi



Slika 2.11: Usmjereni aciklički graf. Vjerojatnosti koje trebamo odrediti su: $P(A)$, $P(B)$, $P(C|A, B)$, $P(E|C)$, $P(D|C)$, $P(F|E)$, i $P(G|D, E, F)$.

	$G = da$	$G = ne$
$P(G)$	0.98	0.02

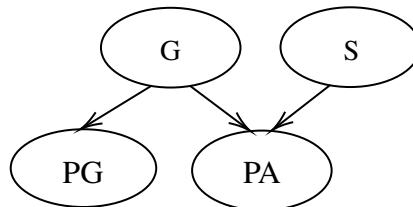
Tablica 2.1: $P(G)$

	$S = da$	$S = ne$
$P(S)$	0.96	0.04

Tablica 2.2: $P(S)$

upaliti. Čuje se početni okretaj, ali ništa se ne događa. Može postojati nekoliko uzroka problema. Mogu čuti pokretanje, tako da akumulator sigurno nije prazan. Dakle, najvjerojatniji uzroci su da je gorivo ukradeno preko noći ili da su svjećice prljave. Može biti i da je prljavština u rasplinjaču, labav spoj u sustavu paljenja ili nešto ozbiljnije. Da saznam, najprije trebam pogledati pokazivač količine goriva. Izgleda da je na pola pun, odlučujem očistiti svjećice.

Kako bi pojednostavili situaciju zamislimo da gledamo događaje s pripadnim stanjima: $\{da, ne\}$ za Gorivo, $\{da, ne\}$ za Čiste svjećice, $\{puno, pola, prazno\}$ za Pokazivač količine goriva i $\{da, ne\}$ za Paljenje auta. Znamo da stanje goriva i stanje čistih svjećica ima kausalni utjecaj na stanje paljenja auta. Također, stanje goriva utječe na stanje pokazivača količine goriva. To je prikazano na slici 2.12. Označimo Gorivo, Čiste svjećice, Pokazivač količine goriva i Paljenje auta respektivno s G , S , PG , PA . Za kvantitativno modeliranje trebamo procjene vjerojatnosti za $P(G)$, $P(S)$, $P(PA|G, S)$, $P(PG|G)$. Tablice s vjerojatnostima su dane respektivno u tablicama 2.1, 2.2, 2.3, 2.4. Uočimo da tablica 2.3 za $P(PA|G, S)$ ostavlja prostor za druge uzroke osim manjka benzina i prljavih svjećica tako da vrijedi $P(PA = ne|G = da, S = da) > 0$, a tablica 2.4 za $P(PG|G)$ ostavlja mogućnost da je mjerač količine goriva pokvaren.



Slika 2.12: Kauzalna mreža za *Problem paljenja auta*. *G* označava *Gorivo*, *S* su *Čiste svjećice*, *PG* je kratica za *Pokazivač količine goriva* te *PA* za *Paljenje auta*.

	G = da	G = ne
S = da	(0.99, 0.01)	(0, 1)
S = ne	(0.01, 0.99)	(0, 1)

Tablica 2.3: Uvjetna vjerojatnost $P(PA|G, S)$. Brojevi (x, y) označavaju ($PA = da, PA = ne$).

	G = da	G = ne
PG = puno	0.39	0.001
PG = pola	0.60	0.001
PG = prazno	0.01	0.998

Tablica 2.4: Uvjetna vjerojatnost $P(PG|G)$.

2.4 Lančano pravilo za Bayesove mreže

Neka je $\mathcal{F} = \{A_1, \dots, A_n\}$ prostor slučajnih varijabli. Ako imamo tablicu združene distribucije vjerojatnosti $P(\mathcal{F}) = P(\{A_1, \dots, A_n\})$, tada možemo izračunati $P(A_i)$, ali i $P(A_i|e)$, gdje je e znanje o nekim varijablama u Bayesovoj mreži. Međutim, $P(\mathcal{F})$ raste eksponencijalno s brojem slučajnih varijabli te već za mali \mathcal{F} tablica postaje prevelika. Stoga nas zanima kompaktniji prikaz $P(\mathcal{F})$.

Neka je BN Bayesova mreža nad \mathcal{F} te neka je $P(\mathcal{F})$ vjerojatnosna distribucija svojstva definiranih u BN :

- uvjetna vjerojatnost varijable uz dane njene roditelje u $P(\mathcal{F})$ mora biti određena u BN ,
- ako su varijable A i B D -separirane u BN na danom skupu C , tada su A i B nezavisne za dani C u $P(\mathcal{F})$.

Možemo zaključiti da ako se \mathcal{F} sastoji od samo jedne varijable A , tada BN definira $P(A)$ te je $P(\mathcal{F})$ jedinstveno definirana. Pokazat ćemo da ovo općenito vrijedi. Za vjerojatnosne distribucije na skupu slučajnih varijabli, koristimo jednadžbu koju nazivamo *lančano pravilo*. Za Bayesove mreže ova jednadžba ima poseban oblik. Prvo ćemo navesti opći oblik jednadžbe.

Propozicija 2.4.1. (Poopćeno lančano pravilo) Neka je $\mathcal{F} = \{A_1, \dots, A_n\}$ skup varijabli. Tada za bilo koju vjerojatnosnu distribuciju $P(\mathcal{F})$ imamo

$$P(\mathcal{F}) = P(A_n|A_1, \dots, A_{n-1})P(A_{n-1}|A_1, \dots, A_{n-2}) \dots P(A_2|A_1)P(A_1).$$

Dokaz. Dokaz slijedi indukcijom iz (2.2). □

Teorem 2.4.2. (Lančano pravilo za Bayesove mreže) Neka je BN Bayesova mreža nad $\mathcal{F} = \{A_1, \dots, A_n\}$. Tada BN određuje jedinstvenu združenu distribuciju vjerojatnosti $P(\mathcal{F})$ danu produktom svih tablica uvjetnih vjerojatnosti definiranih u BN :

$$P(\mathcal{F}) = \prod_{i=1}^n P(A_i|pa(A_i)), \quad (2.5)$$

gdje $pa(A_i)$ označava roditelja čvora A_i u BN , a $P(\mathcal{F})$ odražava svojstva BN .

Lančano pravilo nam govori da je Bayesova mreža kompaktna reprezentacija združene distribucije vjerojatnosti.

	PG = puno	PG = pola	PG = prazno
S = da	(0.363, 0)	(0.599, 0)	(0.0093, 0)
S = ne	(0.00015, 0)	(0.00024, 0)	(3.9·10 ⁻⁶ , 0)

Tablica 2.5: Tablica združene vjerojatnosti za $P(G, PG, S, PA = da)$. Brojevi (x, y) u tablici označavaju $(G = da, G = ne)$.

	PG = puno	PG = pola	PG = prazno
S = da	(0.00367, 1.9·10 ⁻⁵)	(0.00564, 1.9·10 ⁻⁵)	(9.4·10 ⁻⁵ , 0.0192)
S = ne	(0.01514, 8·10 ⁻⁷)	(0.0233, 8·10 ⁻⁷)	(0.000388, 0.000798)

Tablica 2.6: Tablica združene vjerojatnosti za $P(G, PG, S, PA = ne)$. Brojevi (x, y) u tablici označavaju $(G = da, G = ne)$.

Primjer 2.4.3. Ponovno proučimo primjer 2.3.2, "Paljenje auta". Za zaključivanje ćemo koristiti tablicu združene vjerojatnosti. Nju ćemo odrediti pomoću lančanog pravila za Bayesove mreže, (2.5), $P(G, PG, S, PA) = P(G)P(S)P(PG|G)P(PA|G, S)$. Rezultat je dan u tablicama 2.5 i 2.6.

Marginalizacijom varijabli PG i G iz tablice 2.6, odnosno zbrajajući svaki redak, dobivamo: $P(S, PA = ne) = (0.02864, 0.03965)$. Uvjetnu vjerojatnost $P(S|PA = ne)$ dobijemo dijeleći je s $P(PA = ne)$. Kako je

$$P(PA = ne) = P(S = da, PA = ne) + P(S = ne, PA = ne) = 0.02864 + 0.03965 = 0.06829$$

te slijedi

$$P(S|PA = ne) = \left(\frac{0.02864}{0.06829}, \frac{0.03965}{0.06829} \right) = (0.42, 0.58).$$

Ako rezultat ne bude uređeni par brojeva čija je suma 1, tada normaliziramo dijeleći sa sumom.

Na analogan način dobivamo $P(G|PA = ne) = (0.71, 0.29)$. Zatim dobijemo informaciju da je $PG = pola$ te se računanje svodi na $PG = pola$ i $PA = ne$. Rezultat je dan u tablici 2.7.

Marginalizacijom S te normalizacijom, dobijemo $P(G|PA = ne, PG = pola) = (0.999, 0.001)$, zatim marginalizacijom G i normalizacijom slijedi $P(S|PA = ne, PG = pola) = (0.196, 0.804)$. Vjerojatnost da je $S = da$ se povećala uz tvrdnju $PG = pola$, stoga možemo uočiti efekt objašnjava.

	G = da	G = ne
S = da	0.00564	$1.9 \cdot 10^{-5}$
S = ne	0.0233	$8 \cdot 10^{-7}$

Tablica 2.7: $P(G, S, PA = ne, PG = pola)$.

2.5 Dodavanje znanja

Bayesove mreže se koriste za računanje vjerojatnosti kada dobijemo neku novu informaciju, npr. $A = a$, gdje je A varijabla, a a je njeno stanje. Neka A ima n stanja i $P(A) = (x_1, \dots, x_n)$, pretpostavimo da smo dobili novu informaciju e da A može biti samo u stanju i ili j . To znači da su sva stanja osim i i j nemoguća te $P(A, e) = (0, \dots, 0, x_i, 0, \dots, 0, x_j, 0, \dots, 0)$.

Definicija 2.5.1. Neka je A varijabla s n stanja. Nalaz o A je n -dimenzionalna tablica nula i jedinica.

Nalaz e je tvrdnja da su određena stanja od A nemoguća.

Pretpostavimo da imamo tablicu združene vjerojatnosti, $P(\mathcal{F})$ te neka je e nalaz. Tablica združene vjerojatnosti $P(\mathcal{F}, e)$ je tablica dobivena iz $P(\mathcal{F})$ zamjenjujući sva stanja u kojima A nije u stanju i i j s nulom, a ostale ostavljajući nepromijenjenim. To je jednako tome da smo $P(\mathcal{F})$ pomnožili s e ,

$$P(\mathcal{F}, e) = P(\mathcal{F}) \cdot e.$$

Koristeći lančano pravilo za Bayesove mreže, (2.5), imamo sljedeći teorem.

Teorem 2.5.2. Neka je BN Bayesova mreža na prostoru \mathcal{F} te neka su e_1, \dots, e_n tvrdnje da su određena stanja od A nemoguća. Tada

$$P(\mathcal{F}, e) = \prod_{A \in \mathcal{F}} P(A|pa(A)) \cdot \prod_{i=1}^n e_i,$$

te za $A \in \mathcal{F}$ imamo

$$P(A|e) = \frac{\sum_{\mathcal{F} \setminus \{A\}} P(\mathcal{F}, e)}{P(e)}.$$

Poglavlje 3

Prepoznavanje i praćenje objekata

Proučit ćemo praćenje objekata s pogleda Bayesovih mreža. Uz znanje iz prethodnih poglavlja, cilj je definirati i približiti neke načine rješavanja problema praćenja objekata. Započet ćemo objašnjavanjem problema praćenja objekata, a zatim prikazati kako zaključivanjem u Bayesovim mrežama možemo proučiti taj problem.

3.1 Problem praćenja objekata

Problem praćenja objekata uključuje procjenu puta nekog objekta. Pomoću raznih sprava radimo mjerenja pozicije objekta u diskretnom vremenu, $t = t_0 < t_1 < \dots < t_K$ te radi jednostavnosti uzimamo $t_0 = 0$. Ukoliko prilikom mjerenja pozicije ne bi došlo do nikakvih grešaka, mogli bi neometano procjenjivati put objekta. Međutim, tijekom mjerenja nastaju mnoge pogreške koje dovode do problema praćenja objekta.

Pretpostavimo da su svi podaci koje dobivamo upravo oni od samog objekta kojeg pratimo, a ne na primjer uzrok neke greške kao što je šum na podacima. Koristeći te podatke, želimo odrediti položaj praćenog objekta u trenutnom vremenu $t = t_K$, put objekta od početnog vremena do trenutnog vremena: $[0, t_K]$ ili poziciju objekta u nekom vremenu $t > t_K$.

Procjenu trenutne pozicije objekta nazivamo *filtriranje*, procjenu buduće pozicije nazivamo *predikcija*, a *zaglađivanje* označava procjenu prošle pozicije ili prošlog puta. Mi želimo procjenjivati i ostale karakteristike objekta, kao što je brzina ili ubrzanje, stoga, definiramo *stanje* objekta kao vektor komponenata koje želimo procjenjivati. Proces procjene stanja objekata nazivamo *praćenje*. Dakle, cilj praćenja objekata je procjena stanja praćenih objekata sekvencijalno kroz vrijeme. Neki primjeri praćenja objekata su nadzorne kamere te nadziranje tla, navođenje raketa, procjena putanje vozila te ostali problemi koji uključuju praćenje ili vremensku procjenu promatranih stanja.

Vremenski varijabilno stanje objekata se procjenjuje pomoću senzora kao što su radar, sonar, kamera, teleskop. Procjena objekata na temelju senzornih promatranja zahtijeva neku vrstu pratećeg sustava kojeg nazivamo *filter*. Zadaća filtera je kontinuirana procjena stanja praćenog objekta, odnosno, više njih. U vidu Bayesovih mreža, koristeći prethodne informacije procjenjujemo trenutno stanje. Praćenje je problem statističke procjene te se smatra da se taj problem može najbolje riješiti u vidu zaključivanja u Bayesovim mrežama.

3.2 Modeli kretanja

Kako bi proveli praćenje objekata, trebamo matematičke modele za mjerenje greške te model za opisivanje kretanja ciljnog objekta. Model za mjerenje greške je dan u vidu distribucije vjerojatnosti za grešku u mjerenju. Jednostavan primjer modela za kretanje je model konstantne brzine. Za taj model možemo odrediti distribuciju vjerojatnosti za poziciju objekta u vremenu $t = 0$ te distribuciju za brzinu objekta. Ako bi početna pozicija i brzina objekta bili poznati, tada bi njen kompletni put bio poznat. Općenitije, ciljani model gibanja je stohastički proces koji u vjerojatnosnom smislu opisuje kako se ciljani objekt kreće kroz svoja stanja u danom vremenu t .

3.3 Procjena puta objekata

Kod praćenja pojedinog objekta želimo kombinirati izmjerene podatke s mjerenjima i modelima kretanja kako bi procijenili put kojim će se objekt kretati. Točnije, želimo dobiti optimalnu ili najbolju procjenu. To bi značilo da želimo smanjiti očekivanu kvadratnu pogrešku između procijenjenoga i stvarnog puta objekta. Zatim, mogli bismo naći put koji ima najveću vjerojatnost da je točan ili možemo izračunati uvjetnu distribuciju na stanje objekta uz dana mjerenja. U ovom slučaju, početna vjerojatnost bi bili stohastički procesi modeliranja kretanja objekta, a mjerenja su podaci koji nam omogućuju izračunavanje uvjetne vjerojatnosti. U ovom pogledu, pojam zaključivanja u Bayesovim mrežama označava izračunavanje uvjetne distribucije i procjene izvedene iz nje kao što je srednja vrijednost ili maksimum uvjetne vjerojatnosti stanja. Srednja vrijednost je procjena koja minimizira očekivanu kvadratnu grešku između procijenjenog i stvarnog ciljnog puta.

3.4 Bayesovo filtriranje i filteri

Filtriranje, u kontekstu stohastičkog procesiranja signala, ima za cilj izvući najbolju moguću procjenu za stvarno stanje ciljnog objekta koristeći šumovita promatranja objekata. Drugim riječima, filtriranje je matematička operacija koja uključuje izvlačenje informacije o objektu u nekom vremenu t_K , koristeći senzorna promatranja dobivena prije

vremena t_K . Praćeni objekt je najčešće skriven, odnosno ne znamo točne podatke o stanju tog objekta. Bayesovo filtriranje olakšava vjerojatnosno modeliranje ove nesigurnosti koristeći prethodno znanje i dokaze senzora o stanju objekta. Tijekom godina, nastalo je nekoliko Bayesovih filtera kako bi se pokrio što veći raspon problema praćenja objekata. Sada ćemo ukratko spomenuti svaki od njih.

Kalmanov filter

Kalmanov filter je najpoznatiji i najčešći način rješavanja problema praćenja na linearnim dinamičkim sustavima u diskretnim vremenskim trenucima koristeći funkciju prijenosa stanja što je opširnije opisano u 4.1. Predstavio ga je Rudolf Emil Kálmán¹, 1960. Pruža analitičko rješenje za procjenu stanja praćenog objekta na način koji minimizira pogrešku povezanu s odabirom pretpostavke o stanju objekta, stoga je ovaj filter teoretski optimalan. Kalmanov filter se primjenjuje na situacijama gdje je proces kretanja objekata Markovljev ili Gaussov. U slučaju Gaussovog procesa, distribucija stanja objekata u bilo kojem vremenu t je Gaussova, a vektorska distribucija stanja objekata u bilo kojem konačnom skupu vremena je vektorska Gaussova. Dodatno, Kalmanov filter zahtijeva da su mjerenja, linearne funkcije stanja objekata, u vremenu mjerenja s dodanim Gaussovima greškama. Ovaj skup pretpostavka nazivamo *linearne-Gaussove* pretpostavke.

U linearnom Gaussovom slučaju, uvjetna distribucija na stanje objekta u bilo kojem vremenu t je Gaussova. Ona je okarakteriziran srednjom vrijednošću i kovariancom. Kalmanov filter pruža jednostavnu i učinkovitu rekurzivnu funkciju za izračunavanje srednje vrijednosti i kovarijance Gaussove uvjetne distribucije stanja objekta u trenutnom vremenu t_K uz dana mjerenja u t_K i srednje vrijednosti te kovarijancu uvjetne distribucije u vremenu t_{K-1} . U ovom slučaju, srednja vrijednost rješenja Kalmanovog filtera minimizira očekivanu kvadratnu grešku između procijenjenoga i stvarnog ciljnog puta te je maksimalna uvjetna vjerojatnost procjene stanja objekta. U linearnom Gaussovom slučaju, Kalmanov filter nalazi uvjetnu distribuciju na stanje objekta te računa srednju vrijednost distribucije koja minimizira očekivanu grešku te maksimizira uvjetnu vjerojatnost.

Ako zadržimo samo pretpostavke o linearnosti i neovisnosti, Kalmanov filter pruža optimalnu procjenu stanja praćenog objekta koje je linearna funkcija mjerenja. Ova procjena je optimalna u vidu minimizacije očekivane kvadratne greške.

Linearni Kalmanov filter može raditi u stvarnom vremenu (*eng. Real-time*), koristeći na ulazu samo mjerenje i prethodno izračunato stanje, a na izlazu može estimirati i vrijednosti koje nisu mjerene.

U slučaju nelinearnog sustava, koristimo prošireni Kalmanov filter (*eng. Extended Kalman Filter*).

¹ američki matematičar i elektroničar mađarskoga podrijetla, (1930. - 2016.).

Kalmanov filter se koristi u mnogim područjima kao što su navođenje, navigacija i kontrola raznih objekata. Također, često se koristi u raznim procesiranjima signala i ekonometriji.

Filter čestica

Za nelinerano i ne Gaussovo filtriranje, filter čestica koji još i nazivamo *sekvencijalni Monte Carlo* filter, formalno razvijen 1993.², pruža rigoroznu Bayesovu okosnicu za procjenu stanja praćenog objekta. Filter čestica djeluje na principu aproksimacije funkcije distribucije uvjetne vjerojatnosti skupa primjera, koje nazivamo *čestice*.

Filter hipoteze gustoće vjerojatnosti

Filter hipoteze gustoće vjerojatnost (*eng. The probability hypothesis density filter*) razvio je Ronald P. S. Mahler, 2003. godine za praćenje više objekata. [15] Ovaj filter aproksimira funkciju distribucije uvjetne vjerojatnosti na temelju prvog trenutka. Primarno je razvijen da bi prevladao nedostatke konvencionalno optimalnog Bayesovog združenog filtera.

Multi-Bernoulijev filter

Multiobjektni multi-Bernoulijev (*MeMBeR*) filter, razvijen je 2009. ³ Temelji se na teoriji na kojoj se temelji filter hipoteze gustoće vjerojatnosti te djeluje aproksimacijom uvjetne vjerojatnosti skupa multi-Bernoulijevih parametara. Svi filteri osim Kalmanovog su suboptimalni stoga ne daju točan rezultat.

²Gordon, N.J., Salmond, D.J., Smith, A.F.M. u radu [6].

³Vo, Ba Tuong and Vo, Ba-Ngu and Cantoni, Antonio u radu [35].

Poglavlje 4

Prepoznavanje i praćenje lica

U ovom poglavlju ćemo prikazati primjenu Kalmanovog filtera na problemu prepoznavanja i praćenja lica. Kalmanov filter smo uveli u prethodnom poglavlju, ali ćemo sada pobliže objasniti njegovu primjenu. Potom ćemo prikazati kako ga primijeniti na stvarnom problemu detekcije lica.

4.1 Modeliranje prostora stanja praćenog objekta

Pristup modeliranju objekta pomoću prostora stanja uključuje stanje objekta. Stanje obuhvaća potpune informacije o objektu u svom prostoru te se prikazuje pomoću vektora. Prostor stanja je skup svih stanja praćenog objekta. Pretpostavimo da je x_K vektor stanja objekta u vremenu t_K . Ovaj vektor stanja modeliramo kao slučajnu varijablu. Na primjer, za neki objekt koji se giba, neka je p pozicija, a v brzina objekta. Vektor stanja prikazujemo kao $x_K = (p, v)$. Ukoliko postoji još neki senzor, on nam može govoriti nešto o stanju objekta, ali samo indirektno te s nekom nesigurnošću i netočnošću. Naša predikcija nam kaže nešto o kretanju objekta, ali također indirektno s nesigurnošću i netočnošću.

Princip rada Kalmanovog filtera

Ukoliko uzmemo sve informacije koje su nam dane i primijenimo Kalmanov filter možemo dobiti bolji odgovor od samo naše procjene. Mi ne znamo koja su stvarna pozicija i brzina objekta, ali znamo da su neka buduća stanja vjerojatnija od drugih. Kalmanov filter pretpostavlja da su, u ovom slučaju, pozicija i brzina slučajne i Gausovski distribuirane. Svaka varijabla ima svoju srednju vrijednost μ , koja je središte slučajne distribucije te je vrlo vjerojatno i stanje te ima i svoju varijancu σ^2 . Pozicija i brzina mogu biti nekoorelirane, što znači da stanje jedne varijable nam ne govori ništa o tome kakvo bi stanje druge varijable moglo biti. Ukoliko su pozicija i brzina koorelirane, vjerojatnost zadržavanja određene po-

zicije ovisi o brzini koju objekt ima. Ovaj slučaj bi se mogao dogoditi ukoliko, na primjer, procjenjujemo novu poziciju na temelju stare. Ako je brzina velika, vrlo je vjerojatno da se objekt pomaknuo dalje te će pozicija biti dalja, a ukoliko je brzina mala, objekt se nije daleko pomaknuo. Ovakav odnos brzine i pozicije nam daje najviše informacija. Jedno mjerenje nam govori nešto o tome kakvo bi sljedeće moglo biti. To je upravo i cilj Kalmanovog filtera. Korelaciju zapisujemo pomoću matrice korelacije. Svaki element matrice Σ_{ij} je stupanj korelacije između i -tog i j -tog stanja. Za modeliranje znanja trebamo dvije informacije u vremenu t_K . Našu najbolju procjenu \hat{x}_K (srednja vrijednost, inače označena s μ) te matricu kovarijance P_K . U našem primjeru s pozicijom i brzinom, imamo:

$$\hat{x}_K = \begin{bmatrix} \text{pozicija} \\ \text{brzina} \end{bmatrix} \quad (4.1)$$

$$P_K = \begin{bmatrix} \Sigma_{pp} & \Sigma_{pv} \\ \Sigma_{vp} & \Sigma_{vv} \end{bmatrix} \quad (4.2)$$

Zatim, trebamo pogledati trenutno stanje koje znamo, a to je ono u vremenu t_{K-1} i predvidjeti sljedeće stanje u vremenu t_K . Funkcija predikcije nam daje novu distribuciju. Ovo predviđeno stanje možemo označiti matricom F_K . Ona uzima svaku točku naše originalne procjene te je seli na novo predviđeno mjesto, gdje bi se objekt nalazio da je naša originalna predikcija bila točna. Kako bismo predvidjeli poziciju i brzinu objekta u sljedećem trenutku, koristimo sljedeću formulu:

$$\begin{aligned} p_K &= p_{K-1} + \Delta t \cdot v_{K-1} \\ v_K &= v_{K-1} \end{aligned}$$

odnosno, matrica predikcije:

$$\hat{x}_K = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{x}_{K-1} \quad (4.3)$$

$$= F_K \cdot \hat{x}_{K-1} \quad (4.4)$$

Matrica predikcije nam dalje sljedeće stanje, ali još uvijek moramo ažurirati matricu kovarijance. Znamo da ukoliko izmnožimo svaku točku distribucije sa matricom A , matrica kovarijance daje identitetu:

$$\begin{aligned} \text{Cov}(x) &= \Sigma \\ \text{Cov}(Ax) &= A\Sigma A^\tau \end{aligned} \quad (4.5)$$

Iz (4.4) i (4.5) slijedi:

$$\begin{aligned} \hat{x}_K &= F_K \hat{x}_{K-1} \\ P_K &= F_K P_{K-1} F_K^\tau \end{aligned} \quad (4.6)$$

Vanjski utjecaj

Mogu se dogoditi promjene koje nisu vezane uz samo stanje objekta, takve promjene su pod vanjskim utjecajem. Ako znamo informaciju o vanjskim utjecajima, možemo je dodati našoj predikciji kao korekciju. Vanjsku informaciju označavamo vektorom u_K . Pretpostavimo da znamo očekivanu akceleraciju a . Znamo da vrijedi:

$$\begin{aligned} p_K &= p_{K-1} + \Delta t v_{K-1} + \frac{1}{2} a \Delta t^2 \\ v_K &= v_{K-1} + a \Delta t \end{aligned}$$

u matričnom obliku:

$$\begin{aligned} \hat{x}_K &= F_K \hat{x}_{K-1} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a \\ &= F_K \hat{x}_{K-1} + B_K u_k \end{aligned} \quad (4.7)$$

B_K nazivamo kontrolnom matricom, a u_K kontrolnim vektorom.

Vanjska nesigurnost

Ukoliko se stanje mijenja na temelju vanjskih utjecaja, to nije problem ako znamo koji su to utjecaji. Problem nastaje kada utječu vanjski elementi za koje ne znamo. Na primjer da pratimo robota koji se kreće po tlu, a neravnine na tlu ga uspore. Takve stvari ne možemo nužno predvidjeti te naša predikcija zbog toga može biti pogrešna.

Možemo modelirati nesigurnost vezanu za vanjski utjecaj tako da dodamo nove nesigurnosti nakon svakog koraka predikcije. Sve nepoznate vanjske utjecaje tretiramo kao kao šum s kovariancom Q_K . Tada imamo kompletan izraz za korak predikcije:

$$\hat{x}_K = F_K \hat{x}_{K-1} + B_K u_k \quad (4.8)$$

$$P_K = F_K P_{K-1} F_K^T + Q_K \quad (4.9)$$

Drugim riječima, u (4.8) naša nova najbolja procjena \hat{x}_K je suma predikcije F_K dobivene od prethodne najbolje procjene \hat{x}_{K-1} i korekcije B_K za poznate vanjske utjecaje u_K . A u (4.9) nova nesigurnost P_K je predikcija F_K stare nesigurnosti P_{K-1} te dodatne nesigurnosti zbog vanjskih utjecaja Q_K . Sada imamo procjenu gdje bi se naš objekt mogao nalaziti, uz dani \hat{x}_K i P_K .

Redefiniranje procjene s mjerenjima

Postoji mogućnost da postoje senzori koji nam daju dodatne informacije o stanju našeg objekta. Na primjer, pretpostavimo da jedan govori nešto o poziciji, a drugi o brzini

objekta. Dakle, dobivamo indirektne informacije o stanju, odnosno, senzor proizvodi niz očitavanja. Jedinice i mjerilo od očitavanja sa senzora ne moraju odgovarati jedinicama i mjerilu stanja kojeg mi proučavamo. Stoga, modeliramo senzor s matricom H_K . Možemo pretpostaviti distribuciju očitavanja koje smo i očekivali:

$$\begin{aligned}\mu_{\text{očekivano}} &= H_K \hat{x}_K \\ \Sigma_{\text{neočekivano}} &= H_K P_K H_K^T\end{aligned}\quad (4.10)$$

Kalmanov filter je vrlo dobar u rješavanju problema smetnji senzora. Senzori mogu biti nepouzdana i svako stanje u našoj originalnoj procjeni može rezultirati nizom očitavanja senzora. Iz svakog očitavanja koje promatramo, možemo pretpostaviti da je naš objekt bio u određenom stanju. No, budući da postoji nesigurnost, neka stanja su vjerojatnija da su proizvela čitanje koje smo vidjeli. S R_K označimo kovarijancu ove nesigurnosti (na primjer senzorni šum). Distribucija ima srednju vrijednost jednaku očitanju z_K .

Sada imamo srednju vrijednost naše transformirane predikcije i očitavanje senzora. Stoga, moramo pokušati promijeniti naše pretpostavke o očitanjima koja smo vidjeli na temelju predviđenog stanja, s drugačijim nagađanjem zasnovanim na očitanjima senzora koje smo zapravo promatrali.

Postavlja se pitanje koje je najvjerojatnije stanje. Za bilo koje očitavanje (z_1, z_2) imamo dvije vezane vjerojatnosti:

- vjerojatnost da je očitavanje senzora z_K zapravo pogrešno mjerenje (z_1, z_2)
- vjerojatnost da naša prethodna procjena smatra (z_1, z_2) očitanjem koje bismo trebali upotrijebiti.

Imamo dvije vjerojatnosti i zanima nas koja je vjerojatnost da su obje točne. To će biti preciznije od prethodnih procjena. Srednja vrijednost ove distribucije je konfiguracija za koju su obje procjene vrlo vjerojatne što bi bila i najbolja pretpostavka o stvarnoj konfiguraciji uz sve informacije koje imamo. Stoga, trebamo povezati pretpostavke.

Povezivanje pretpostavka

Neka je A neka jednodimenzionalna Gaussova zvonolika krivulja s varijancom σ^2 i srednjom vrijednošću μ , definirana s:

$$\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (4.11)$$

Zanima nas

$$\mathcal{N}(x, \mu_0, \sigma_0) \cdot \mathcal{N}(x, \mu_1, \sigma_1) \stackrel{?}{=} \mathcal{N}(x, \mu', \sigma') \quad (4.12)$$

Substitucijom (4.11) u (4.12) dobijemo:

$$\begin{aligned}\mu' &= \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2} \\ \sigma'^2 &= \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}\end{aligned}\quad (4.13)$$

Označimo

$$k = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \quad (4.14)$$

Substitucijom k u (4.13) dobivamo:

$$\begin{aligned}\mu' &= \mu_0 + k(\mu_1 - \mu_0) \\ \sigma'^2 &= \sigma_0^2 - k\sigma_0^2\end{aligned}\quad (4.15)$$

Zapišimo (4.14) i (4.15) u matičnom zapisu. Ako je Σ matrica kovarijance, a μ srednja vrijednost uzduž osi, tada:

$$K = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1} \quad (4.16)$$

$$\begin{aligned}\mu' &= \mu_0 + K(\mu_1 - \mu_0) \\ \Sigma' &= \Sigma_0 - K\Sigma_0\end{aligned}\quad (4.17)$$

K je matrica koju nazivamo *Kalmanovo pojačanje* (eng. *Kalman gain*).

Konačna estimacija

Sada imamo dvije distribucije: predviđeno mjerenje s $(\mu_0, \Sigma_0) = (H_K \hat{x}_K, H_K P_K H_K^T)$ i opažena mjerenja s $(\mu_1, \Sigma_1) = (z_K, R_K)$. Zajedno sa (4.17) imamo:

$$\begin{aligned}H_K \hat{x}'_K &= H_K \hat{x}_K + K(z_K - H_K \hat{x}_K) \\ H_K P'_K H_K^T &= H_K P_K H_K^T - K H_K P_K H_K^T\end{aligned}\quad (4.18)$$

Iz (4.16), Kalmanovo pojačanje iznosi:

$$K = H_K P_K H_K^T (H_K P_K H_K^T + R_K)^{-1} \quad (4.19)$$

Iz (4.18) i (4.19) dobivamo konačnu jednadžbu za korak ažuriranja:

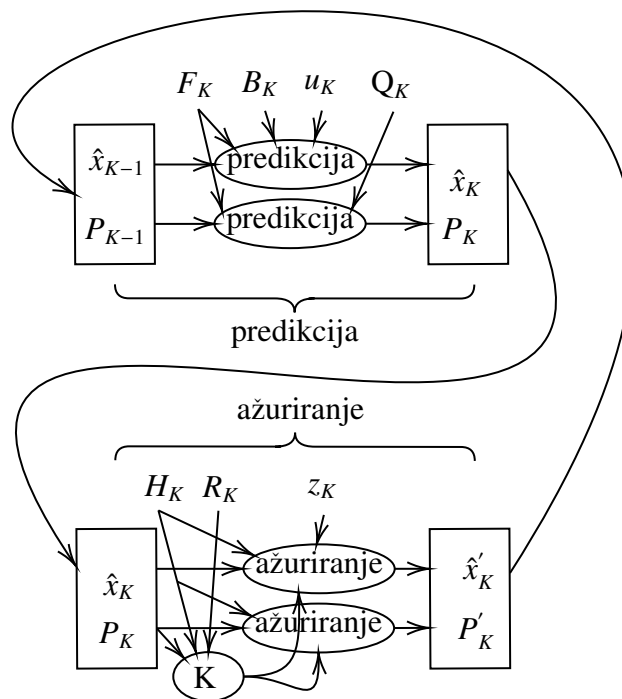
$$\begin{aligned}\hat{x}'_K &= \hat{x}_K + K'(z_K - H_K \hat{x}_K) \\ P'_K &= P_K - K' H_K P_K\end{aligned}\quad (4.20)$$

$$K' = P_K H_K^T (H_K P_K H_K^T + R_K)^{-1} \quad (4.21)$$

Dakle, \hat{x}'_K je naša nova najbolja procjena te zajedno s P'_K rade sljedeću predikciju ili ažuriranje.

Tijek informacija unutar Kalmanovog filtera

Na slici 4.1 prikazan je tijek informacija i dosad objašnjeni proces unutar Kalmanovog filtera.



Slika 4.1: Tijek informacija unutar Kalmanovog filtera.

4.2 Problem prepoznavanja i praćenja lica

Praćenje lica je jedan od najčešće proučavanih tema u području kompjuterskog vida. Male varijacije u veličini i orijentaciji lica mogu utjecati na rezultat praćenja lica.

Lice je zakrivljena trodimenzionalna površina, a u videozapisima može doći do promjena u ljudskom licu zbog osvjetljenja, položaja, frizura, kose na licu, šminke i dobi. Sva lica imaju u osnovi isti oblik, a ipak lice svake osobe je drugačije.

Programsko rješenje za prepoznavanje i praćenje lica

Programsko rješenje za detekciju i praćenje lica napisano je u Python programskom jeziku, koristeći OpenCV biblioteku. Cilj je na temelju ulaznog videa prepoznati i pratiti lice ukoliko ono na videu zaista postoji. Za prepoznavanje lica korištene su istrenirane Haarove kaskade, a za praćenje je korišten Kalmanov filter.

Biblioteka OpenCV

OpenCV - *Open Source Computer Vision Library* je biblioteka otvorenoga koda koja nudi infrastrukturu za razvoj aplikacija u području kompjuterskog vida i strojnog učenja. Biblioteka OpenCV je dizajnirana za razvoj aplikacija u stvarnom vremenu te daje jednostavnu infrastrukturu kojom olakšava njihov razvoj. Biblioteka sadrži optimizirane implementacije 2500 algoritama namijenjenih kompjuterskom vidu, kao što je prepoznavanje i detekcija lica, objekata, prepoznavanje ljudskih pokreta u videu, praćenje objekata te mnoge druge. Napisana je i optimizirana u C/C++ okruženju te uz pomoć OpenCL-a omogućuje rad na heterogenim sustavima. [32]

Ulazni podaci

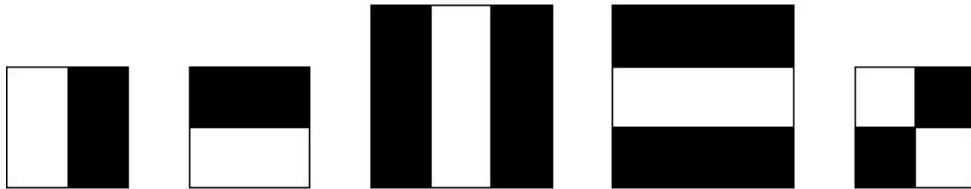
Ulazni podatak je videozapis na kojem se nalazi osoba. Nisu potrebna predprocesiranja slike, osim da je promatramo u sivim tonovima. Detaljnije će biti opisano u području prepoznavanja.

Prepoznavanje

Prepoznavanje lica je postignuto korištenjem metode temeljene na unaprijed istrenirane Haarove kaskade za lice. Ova metoda detekcije je predstavljena od strane Paul Viola i Michael Jones u radu [34]. Metoda Viola-Jones je jedna od prvih metoda koja postiže jako dobre rezultate u stvarnom vremenu. Uz prepoznavanje lica, metoda je primjenjiva za prepoznavanje i drugih objekata na slici. Njena implementacija je integrirana u OpenCV biblioteku. Algoritam ima tri važne značajke koje ćemo detaljnije opisati:

- Korištenje integralne slike koja detektoru omogućuje brzo izračunavanje značajke slike
- Izgradnja klasifikatora - klasifikator je izgrađen korištenjem AdaBoost algoritma za strojno učenje koji izdvaja mali broj kritičnih značajki slike od vrlo velikog broja potencijalnih značajki slike
- Metoda kombinacije klasifikatora u kaskadu - omogućava da se brzo odbace suvišni predjeli slike poput pozadine i time detaljnije provede obrada dijela slike koji sadržava traženi objekt

Inicijalno, algoritam treba mnogo pozitivnih (slike lica) i negativnih slika (slike koje ne sadrže lice) kako bi se istrenirao klasifikator. Stoga trebamo izvući značajke iz tih slika. Za to se koriste Haarove značajke prikazane na slici 4.2. Haarove značajke su pravokutne značajke i računaju se kao razlika sume piksela unutar bijelog područja i sume piksela unutar crnog područja različitih pravokutnih elemenata. Translacijom i skaliranjem osnovnih značajki u prozoru veličine 24×24 se dobiva ukupan skup od otprilike 160000 značajki, što je vrlo zahtjevno ako se treba često ponavljati. Za ubrzanje tog postupka može se koristiti integralna slika. Svaki slikovni element u integralnoj slici S_i dobije se tako da se izračuna zbroj svih slikovnih elemenata originalne slike S lijevo i iznad traženog slikovnog elementa, uključujući i traženi. Primjer takve slike je dan na slici 4.3.



Slika 4.2: Primjer Haarovih značajki. [27]

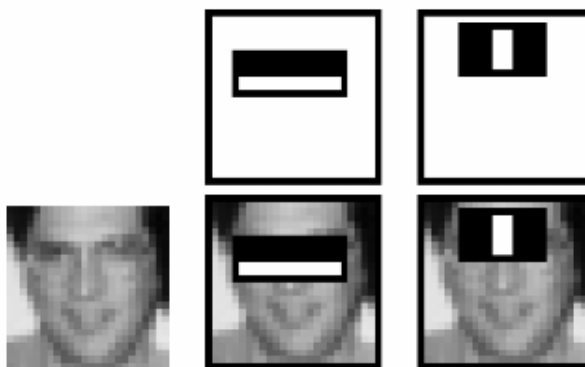
1	1	1
1	1	1
1	1	1

1	2	3
2	4	6
3	6	9

Ulazna slika Integralna slika

Slika 4.3: Reprezentacija slike u integralnom obliku.

Unutar velikog broja značajki koje izračunamo, mnogo ih je irelevantnih. Proučimo sliku 4.4. Gornji red prikazuje dvije dobre značajke. Prva odabrana značajka se oslanja na činjenicu da je regija očiju često tamnija od područja nosa i obraza. Druga odabrana značajka oslanja se na činjenicu da su oči tamnije od hrbta nosa. Ali isti prozori primijenjeni na obraze ili bilo koje drugo mjesto su irelevantni.



Slika 4.4: Primjer dobrih značajki. [31]

Odabir najboljih značajki se postiže koristeći *Adaboost*, što je skraćena za *Adaptive Boosting*. Kako i samo ime kaže radi se o varijanti boosting algoritma. Općenito govoreći boosting algoritmi spajaju slabe klasifikatore u jake klasifikatore iterativno mijenjajući težine pripadnih slabih klasifikatora u procesu učenja. Adaboost uvodi adaptivnu komponentu u proces učenja. Dok je težina, odnosno važnost svih primjera za učenje u klasičnim boosting algoritmima jednaka u svim koracima učenja, Adaboost mijenja težine primjera za učenja. Točnije, u svakom koraku povećava težine, odnosno važnost primjera koji su u prethodnom koraku bili pogrešno klasificirani.

Adaboost je iterativni algoritam koji na osnovu postojećih, možda slabijih, procjenitelja $h_t, t = 1, \dots, T$, konstruira uređeni procjenitelj

$$f(x) = \sum_{t=1}^T \omega_t h_t(x). \quad (4.22)$$

Pri tome se težine ω_t pripadnoga slabog klasifikatora biraju na način da minimiziraju očekivanje greške na testnom skupu podataka. Adaboost algoritam se tipično primjenjuje rekursivno na način da se procjenitelj h_{T+1} dodaje tako da se minimizira po ω_{T+1} . Težine ω_t određujemo tako da je suma greške E_t konačnog klasifikatora, pri treniranju minimizirana:

$$E_t = \sum_i E[f_{t-1}(x_i) + \omega_t h_t(x_i)], \quad (4.23)$$

gdje je f_{t-1} klasifikator konstruiran u prethodnoj fazi treniranja, a E označava neku funkciju greške.

U našem primjeru to znači da se svaka značajka primjenjuje na svim slikama seta za treniranje. Za svaku značajku nalazi se najbolji prag koji će klasificirati lica na pozitivno i negativno. Očito će biti pogrešaka ili pogrešnih klasifikacija. Odabiremo značajke s minimalno pogrešaka, što znači da su izabrane one značajke koje najpreciznije klasificiraju slike lica i slike koje ne prikazuju lica.

Konačni klasifikator je kombinacija tih slabih klasifikatora. Nazivamo ih slabima jer je njihova točnost malo bolja od slučajnog pogađanja te sami ne mogu klasificirati sliku, ali zajedno s drugima (primjenom Adaboost algoritma) tvore snažan klasifikator.

Na slici, veći dio je područje bez lica. Stoga je bolje imati jednostavan način za provjeru da li prozor nije područje lica. Ukoliko zaista nije područje lica, odbacujemo ga i ne obrađujemo više. Umjesto toga, obrada se koncentrira samo na one regije slike gdje je veća vjerojatnost prepoznavanja lica. Na taj način provodimo više vremena provjere mogućih regija lica. To dobivamo konceptom *kaskadnih klasifikatora*. Glavna ideja je složiti klasifikatore u grupe, koje se primjenjuju slijedno jedna iza druge, s tim da odbacivanje primjera od strane jedne grupe (klasifikacija kao pozadina, negativan primjer) prekida daljnji postupak klasifikacije, dok prihvaćanje od strane grupe prosljeđuje primjer sljedećoj grupi. Primjer je pozitivno klasificiran ukoliko je prošao sve grupe u kaskadi. Ovakva struktura osigurava rano odbacivanje primjera koji pripadaju negativnoj klasi te na taj način ubrzava postupak detekcije.

Haarove kaskade čini kaskadni skup Haarovih svojstva. Informacije o Haarovim kaskadama zapisuju se u XML datoteku. U ovom radu su korištene već naučene kaskade za detekciju lica koje su preuzete s OpenCV repozitorija na GitHub-u [33]. Postoje mnoge istrenirane kaskade za različite objekte no ovdje su korištene potrebne istrenirane kaskade za lica.

Prepoznavanje započinjemo uključivanjem potrebnih XML klasifikatora. OpenCV klasa `CascadeClassifier` implementira funkcionalnosti za detekciju objekata na slici. Zatim učitavamo videozapis te uzimamo prvi okvir. Taj okvir otvaramo u sivim tonovima te regiju lica na danom okviru detektiramo pozivom metode `detectMultiScale()`. Odnosno, korištenjem unaprijed istreniranog Viola-Jones detektora.

Praćenje

Sada kada imamo detektirano lice na prvom okviru, za praćenje koristimo Kalmanov filter koji je detaljno opisan u 4.1. Izvodimo sljedećih 5 koraka:

- 1. korak- inicijalizacija početnih vrijednosti, ovaj korak izvodimo samo na početku
- 2. korak- predikcija
- 3. korak- izračunavanje Kalmanovog pojačanja

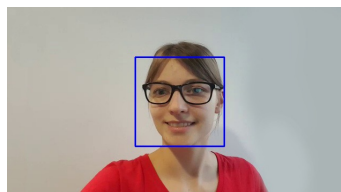
- 4. korak- izračunavanje estimata
- 5. korak- izračunavanje kovarijance greške

Korake 2. do 4. ponavljamo za svaki okvir videozapisa. OpenCV klasa `KalmanFilter` implementira funkcionalnosti standardnog Kalmanovog filtera, a modificirani su neki javni atributi.

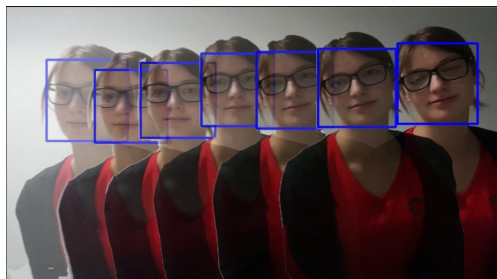
Rezultati i zaključak

Nakon testiranja nekoliko videozapisa praćenja jednog objekta na slici, može se zaključiti da inicijalno prepoznavanje radi dobro, a Kalmanov filter pravilno radi estimaciju pokreta te se objekt kontinuirano prati. Na slici 4.5 vidimo primjer jednog okvira ulaznog videa, a primjer kontinuiranog praćenja vidimo na slici 4.6. Ukoliko osoba okrene lice u suprotnom smjeru i dalje je prati, ali ima trenutaka kada u toj poziciji više ne prati lice ili ga označi premalim okvirom kao što vidimo na slici 4.7. Ukoliko ovaj model primijenimo na prepoznavanje i praćenje više objekata, detekcijom opet odredimo inicijalne pozicije svih osoba, a ukoliko osoba prekrije lice, više je neće pratiti. Što je i bilo za očekivati obzirom da je korišten prepoznavatelj lica. Primjere rezultata vidimo na slici 4.8.

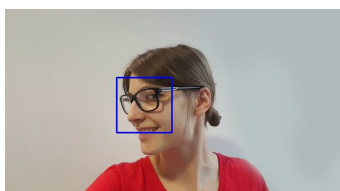
Algoritam za prepoznavanje i praćenje lica je uspješno primijenjen na ulaznim podacima. Rezultati pokazuju da je algoritam uspješno prepoznao i pratio jedno i više lica na videozapisima.



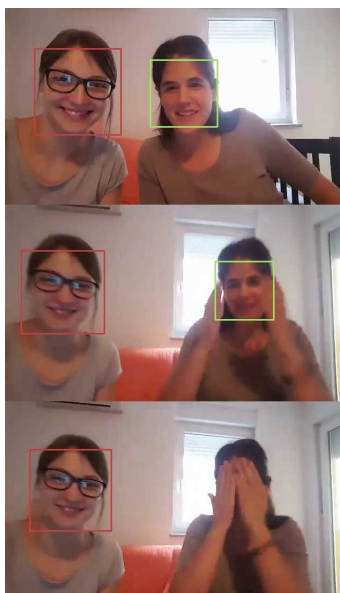
Slika 4.5: Primjer jednog prepoznatog okvira.



Slika 4.6: Primjer kontinuiranog praćenja.



Slika 4.7: Prepoznavanje i praćenje profila lica.



Slika 4.8: Prepoznavanje i praćenje više osoba.

Bibliografija

- [1] Steen Andreassen, Finn V Jensen, Stig Kjær Andersen, B Falck, U Kjærulff, M Woldbye, AR Sørensen, A Rosenfalck i F Jensen, *MUNIN: an expert EMG Assistant*, Computer-aided electromyography and expert systems, Pergamon Press, 1989, str. 255–277.
- [2] Tim Babb, *How a Kalman filter works, in pictures*, <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/#mjax-eqn-kalupdatefull>, (kolovoz 2017.).
- [3] Irad Ben-Gal, *Bayesian networks*, Encyclopedia of statistics in quality and reliability **1** (2008).
- [4] Praveen Babu Choppala, *Bayesian multiple target tracking*, (2014).
- [5] Ian Goodfellow, Yoshua Bengio i Aaron Courville, *Deep learning*, sv. 1, MIT press Cambridge, 2016.
- [6] Neil J Gordon, David J Salmond i Adrian FM Smith, *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, IEE Proceedings F (Radar and Signal Processing), sv. 140, IET, 1993, str. 107–113.
- [7] David Heckerman, *A tutorial on learning with Bayesian networks*, Learning in graphical models, Springer, 1998, str. 301–354.
- [8] ED Heckerman i NB Nathwani, *Toward normative expert systems: part II probability-based representations for efficient knowledge acquisition and inference*, Methods of information in Medicine **31** (1992), br. 02, 106–116.
- [9] Dragana Jankov Maširević i Matea Klarić, *Primjena Bayesove formule i algoritamskog pristupa Bayesovoj formuli na situacijama iz svakodnevnog života*, Math. e **31** (2017), br. 1, 23–33, (kolovoz 2017.).
- [10] Michael Irwin Jordan, *Learning in graphical models*, sv. 89, Springer Science & Business Media, 1998.

- [11] Pedro M Jorge, Arnaldo J Abrantes i Jorge S Marques, *On-line object tracking with Bayesian networks*, IEEE ICIP, 2002.
- [12] P Jorge, A Abrantes i J Marques, *On-line tracking groups of pedestrians with bayesian networks*, Workshop PETS, Prague, Czech Republic, 2004.
- [13] Daphne Koller i Nir Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.
- [14] Krešimir Kukuljan, *Detekcija regija lica na temelju slike kamere mobilnog uređaja*, Disertacija, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Software Engineering. Chair of Programming Languages and Systems., 2017.
- [15] Ronald PS Mahler, *Multitarget Bayes filtering via first-order multitarget moments*, IEEE Transactions on Aerospace and Electronic systems **39** (2003), br. 4, 1152–1178.
- [16] Marvin Minsky, *Steps toward artificial intelligence*, Proceedings of the IRE **49** (1961), br. 1, 8–30.
- [17] Monika Mišolić, *Dodatne teme u srednjoškolskoj nastavi matematike-Teorija grafova*, Disertacija, University of Zagreb. Faculty of Science. Department of Mathematics., 2017.
- [18] Kevin Murphy, *A brief introduction to graphical models and Bayesian networks*, (1998).
- [19] ———, *An introduction to graphical models*, Rap. tech (2001), 1–19.
- [20] Radford M Neal, *Bayesian learning for neural networks*, sv. 118, Springer Science & Business Media, 2012.
- [21] Thomas Dyhre Nielsen i Finn Verner Jensen, *Bayesian networks and decision graphs*, Springer Science & Business Media, 2009.
- [22] Judea Pearl, *Reverend Bayes on inference engines: A distributed hierarchical approach*, Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.
- [23] ———, *Probabilistic reasoning in intelligent systems: Networks of plausible reasoning*, 1988.
- [24] J Pearl i S Russell, *Handbook of Brain Theory and Neural Networks, chapter Bayesian Networks*, Cambridge (2002).

- [25] Franz Pernkopf, Robert Peharz i Sebastian Tschiatschek, *Introduction to probabilistic graphical models*, Academic Press Library in Signal Processing, sv. 1, Elsevier, 2014, str. 989–1064.
- [26] Nikola Sarapa, *Teorija vjerojatnosti*, Školska knjiga, 1987.
- [27] SimplyML, *Face Detection on Your Photo Collection in Python*, <https://simplyml.com/face-detection-on-your-photo-collection-in-python/>, (kolovoz 2017.).
- [28] David J Spiegelhalter i Robin P Knill-Jones, *Statistical and knowledge-based approaches to clinical decision-support systems, with an application in gastroenterology*, Journal of the Royal Statistical Society. Series A (General) (1984), 35–77.
- [29] Timo Stich, *Bayesian networks and structure learning*, Disertacija, Diploma Thesis, Computer Science and Engineering, University of Mannheim, 2004.
- [30] Lawrence D Stone, Roy L Streit, Thomas L Corwin i Kristine L Bell, *Bayesian multiple target tracking*, Artech House, 2013.
- [31] OpenCV Developers Team, *Face Detection using Haar Cascades*, https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html, (kolovoz 2017.).
- [32] ———, *OpenCV*, (kolovoz 2017.).
- [33] ———, *OpenCV*, <https://github.com/opencv/opencv/tree/master/data/haarcascades>, (kolovoz 2017.).
- [34] Paul Viola i Michael Jones, *Rapid object detection using a boosted cascade of simple features*, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, sv. 1, IEEE, 2001, str. I–I.
- [35] Ba Tuong Vo, Ba Ngu Vo i Antonio Cantoni, *The cardinality balanced multi-target multi-Bernoulli filter and its implementations*, IEEE Transactions on Signal Processing **57** (2009), br. 2, 409–423.
- [36] Jan Šnajder, *17. Probabilistički grafički modeli*, https://www.fer.unizg.hr/_download/repository/SU-2017-17-ProbabilistickiGrafickiModeli.pdf, (kolovoz 2017.).

Sažetak

Bayesove mreže važno su područje istraživanja te se koriste za različite primjene u mnogim područjima umjetne inteligencije. U ovom radu opisuje se struktura Bayesovih mreža te se radi kratak pregled vjerojatnosti i statistike zajedno s Bayesovim teoremom. Nadalje uvodimo problem prepoznavanja i praćenja objekata te dajemo pregled Bayesovih filtera. U praktičnom dijelu rada napravljena je primjena algoritma u prepoznavanju i praćenju lica primjenom Viola-Jones algoritma te Kalmanovog filtera.

Summary

Bayesian Networks are an important area of research and are used for various applications within the domain of Artificial Intelligence. This paper describes the structure of Bayesian networks and provides a short overview of probability and statistics together with the Bayesian theorem. Furthermore, we introduce the problem of detection and tracking objects and give an overview of Bayesian filters. In the last chapter we present case studies that use an algorithm for recognizing and tracking faces. The algorithm was made using the Viola-Jones algorithm and Kalman's filter.

Životopis

Rođena sam 12. rujna 1994. godine u Varaždinu. Nakon osnovne škole upisujem Medicinsku školu u Varaždinu te je završavam sa zvanjem Medicinske sestre. Zanimanje za matematiku me potaklo da 2012. godine upišem preddiplomski sveučilišni studij Matematike; smjer: nastavnički, na Prirodoslovno-matematičkom fakultetu u Zagrebu. Godine 2015. upisujem diplomski studij Računarstva i matematike. Tijekom diplomskog studija se zapošljam u tvrtki Photomath gdje i danas radim kao softverski inženjer.