nedeljainformatike v2.0

Probabilističke metode i mašinsko učenje

Osnovni pojmovi verovatnoće

Andrej Ivašković

Matematička gimnazija

Nedelja informatike v2.0

16. decembar 2015.

Zašto nas ovo zanima?

- Najviše istraživanja u oblasti algoritama se tiče *probabilističkih* algoritama!
- Koliko je "dobra" heš funkcija je u sprezi sa verovatnoćom kolizije, raspodelom učestanosti za sva polja...
- Sveprisutan alat u veštačkoj inteligenciji! Aktuelno najbolje rešenje za go.
- Jedna interpretacija talasne funkcije u kvantnoj mehanici je probabilistička – *kvantno računarstvo*?

Prekršeno pravilo

- Prošle godine smo ponavljali kako ne planiramo da držimo predavanja koja zahtevaju napredni matematički aparat. Izgleda da smo to već ove godine prekršili?!
- Čitav današnji dan je svojevrstan "eksperiment".
- Ukoliko se pokaže da koncepti verovatnoće ne predstavljaju barijeru (molimo za detaljne reakcije!), to otvara vrata **mnogim** potencijalnim temama (neke navedene u prethodnom slajdu).

Teme

Šta NEĆEMO pokriti:

- generisanje slučajnih brojeva;
- Monte Karlo algoritmi;
- zakoni velikih brojeva;
- statističke metode ocene parametara (vidi predmet "Verovatnoća i statistika").

Šta ĆEMO pokriti (kroz primere):

- klasična verovatnoća;
- uslovna i totalna verovatnoća;
- Bajesova formula (važno za *sva probabilistička zaključivanja*);
- slučajna promenljiva (važno za *Markovljeve lance*).

Interpretacije verovatnoće

- **Klasična** (Laplasova): odnos broja (odnosno "mere") "traženih" ishoda nekog eksperimenta i broja (ili "mere") švih mogućih" ishoda.
- **Frekventistička**: učestanost nekog događaja "u vremenu".
- **Subjektivna**: "mera poverenja" nekog ishoda.
- **Aksiomatska**: vidi predmet *Verovatnoća i matematička statistika.*

...i mnoge druge!

Osnovni pojmovi

- Uvodimo **prostor događaja** (Ω) kao skup čije elemente (ω) nazivamo **elementarnim događajima**. **Događaj** je bilo koji podskup $A \subseteq \Omega$.
- Bacamo novčić dva puta: $\Omega = \{\mathrm{GG}, \mathrm{GP}, \mathrm{PG}, \mathrm{PP}\}$
- Zanimaciju nas oni ishodi u kojima smo dobili glavu bar jednom: to su slučajevi GG, GP, PG. Neka je $A = \{\mathrm{GG}, \mathrm{GP}, \mathrm{PG}\}$ ovaj događaj.
- Po klasičnoj definiciji, verovatnoća dobijanja bar jedne glave je zato $\frac{3}{4}$.

Algebra događaja

- Dakle, verovatnoća je funkcija $\mathbb{P} : \mathcal{P}(\Omega) \to \mathbb{R}$.
- U prethodnom primeru pišemo $\mathbb{P}(A) = \frac{3}{4}$.
- **Komplement** događaja A je događaj $\overline{A}$: nije dobijena nijedna glava ($\Omega \setminus A$). U ovom primeru: $\mathbb{P}\left(\overline{A}\right) = \frac{1}{4}$.
- Uvek je $\mathbb{P}(A) + \mathbb{P}\left(\overline{A}\right) = 1$.
- Ako posmatramo više događaja, možemo da razmatramo i njihove **preseke** i **unije**.
- Ako je B događaj dobijanja bar jednog pisma, $\mathbb{P}(A \cup B) = 1$, $\mathbb{P}(A \cap B) = \frac{1}{2}$.
- Važi **formula uključenja-isključenja**:

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$$

Nezavisni događaji

- Alisa baca novčić jednom, a Bob baca kockicu za igru. Koja je verovatnoće da su dobijeni glava i neki od brojeva $1, 2, 3, 5$.
- A: Alisa je dobila glavu; B: Bob je dobio $1, 2, 3, 5$.
- Možemo detaljnim ispisom da ustanovimo $\mathbb{P}\left(A \cap B\right) = \frac{1}{3}$. Primetimo $\mathbb{P}\left(A \cap B\right) = \mathbb{P}\left(A\right)\mathbb{P}\left(B\right)$, događaji A i B su **nezavisni**.
- Kada znamo da su neki događaji nezavisni?

Dalje od klasične definicije

- Umesto "klasične interpretacije", događajima ω_i pridružujemo brojeve p_i koji zadovoljavaju $0 \leqslant p_i \leqslant 1$ i još nekoliko intuitivnih uslova (koji su to uslovi?).
- Alisa ima novčić za koji se zna da se glava dobija sa verovatnoćom 0.6. Baca ga dva puta.
- A: prvi put je dobijena glava; B: drugi put je dobijena glava.
- A i B su nezavisni, pa je $\mathbb{P}(A \cap B) = 0.36$. Stoga $\mathbb{P}(A \cup B) = 0.64$.
- Međutim, $A \cup B$ i A *nisu* nezavisni: $A \cap (A \cup B) = A$, a očigledno $\mathbb{P}(A) \neq \mathbb{P}(A)\,\mathbb{P}(A \cup B)$.
- Dakle, moramo da uvedemo neki pojam koji opisuje zavisnost verovatnoće jednog događaja od drugog.

Definicija uslovne verovatnoće

- Uvodimo oznaku $\mathbb{P}(A \mid B)$ za "verovatnoća da će se desiti A ako se desilo B".
- Primer (klasičan): "ako sam u dva bacanja novčića dobio bar jednu glavu, koja je verovatnoća da sam dobio i bar jedno pismo?" Razmatranjem slučajeva vidimo $\frac{2}{3}$.
- Neka A: bar jedno pismo; B: bar jedna glava. Vidimo $\mathbb{P}(A) = \mathbb{P}(B) = \frac{3}{4}$.
- $A \cap B$: bar jedna glava i bar jedno pismo, $\mathbb{P}(A \cap B) = \frac{1}{2}$.
- Stoga je prirodno definisati **uslovnu verovatnoću**:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

- Za nezavisne A i B je $\mathbb{P}(A \mid B) = \mathbb{P}(A)$.

Totalna verovatnoća

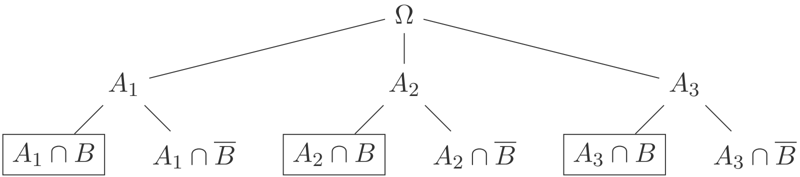
Verovatnoća da će pitanje iz Operativnih sistema na ispitu biti lako je 0.2, a u tom slučaju je verovatnoća da ću ga uraditi jednaka 0.9. Verovatnoća da će pitanje biti srednje težine je 0.5, a uradiću ga sa verovatnoćom 0.7. Ako je pitanje teško, verovatnoća da ću ga uraditi je 0.3. Ako ne znam unapred koja je težina pitanja, koja je verovatnoća da ću ga uraditi?

- A_1, A_2, A_3: događaji lakog, srednjeg, teškog pitanja, respektivno. Ovi skupovi čine jednu **particiju** Ω.
- B: uspešno sam uradio pitanje.
- $\mathbb{P}(A_1) = 0.2$, $\mathbb{P}(A_2) = 0.5$, $\mathbb{P}(A_3) = 0.3$, $\mathbb{P}(B \mid A_1) = 0.9$, $\mathbb{P}(B \mid A_2) = 0.7$, $\mathbb{P}(B \mid A_3) = 0.3$.

Totalna verovatnoća (*cont*)

- Tada je **totalna verovatnoća** da ću uraditi pitanje jednaka:

$$
\begin{aligned}
\mathbb{P}(B) &= \mathbb{P}(A_1)\,\mathbb{P}(B \mid A_1) \\
&+ \mathbb{P}(A_2)\,\mathbb{P}(B \mid A_2) \\
&+ \mathbb{P}(A_3)\,\mathbb{P}(B \mid A_3) \\
&= 0.62
\end{aligned}
$$

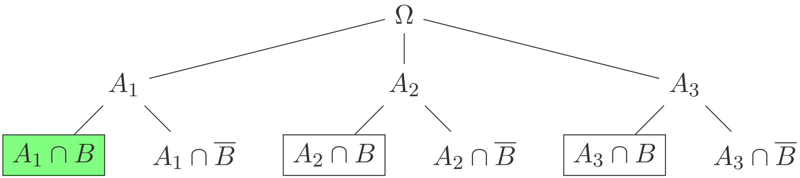


Bajesova formula

Uradio sam pitanje iz OS. Koja je verovatnoća da je ono bilo lako?

- Sada nam je B skup mogućih ishoda, a interesuje nas $\mathbb{P}(A_1 \mid B)$:

$$\mathbb{P}(A_1 \mid B) = \frac{\mathbb{P}(A_1 \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(A_1)\,\mathbb{P}(B \mid A_1)}{\mathbb{P}(B)} \approx 0.29$$



Bajesova formula (*cont*)

- Setimo se izraza za $\mathbb{P}(B)$. Tada je, u stvari, $\mathbb{P}(A_1)\,\mathbb{P}(B \mid A_1)$ đoprinos" u izrazu za $\mathbb{P}(B)$ koji dolazi od $\mathbb{P}(A_1)$.

Bajesova formula

Ako su X i Y neki događaji, tada važi:

$$\mathbb{P}(Y \mid X) = \frac{\mathbb{P}(Y)\,\mathbb{P}(X \mid Y)}{\mathbb{P}(X)}$$

- Ovo je *veoma važan* rezultat!
- Način da se upamti je prosto razumevanje toga šta se, u stvari, traži.
- Izvođenje je vrlo jednostavno – samo iskoristiti definiciju uslovne verovatnoće!

Diskretna slučajna promenljiva

- Možemo da zamislimo jedan RNG (*random number generator*) koji određuje koja će biti vrednost neke promenljive X.
- Stoga možemo da govorimo o verovatnoći da promenljiva uzima neku vrednost.

Kockica nije ravnomerna, pa se zna da se u $\frac{7}{36}$ slučajeva dobije šestica, u $\frac{5}{36}$ slučajeva jedinica, a dvojka, trojka, četvorka i petica sa verovatnoćom $\frac{1}{6}$.

- Neka je X **slučajna promenljiva** koja predstavlja rezultat bacanja ove kockice.
- Tada vidimo: $\mathbb{P}(X = 2) = \frac{1}{6}$; $\mathbb{P}(X = 1) = \frac{5}{36}$; $\mathbb{P}(X = 9) = 0$.
- Može da se piše i:

$$X : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \frac{5}{36} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{7}{36} \end{pmatrix}$$

Diskretna slučajna promenljiva (*cont*)

- Prirodno je govoriti o **raspodeli verovatnoća** p neke promenljive X. Tada je p funkcija $p : \mathbb{R} \to \mathbb{R}$ definisana sa:
$$p(x) = \mathbb{P}(X = x)$$
- Reč je o diskretnoj promenljivoj jer je skup dopuštenih vrednosti diskretan (konačan ili *prebrojivo* beskonačan).
- Definišemo i **kumulativnu funkciju raspodele** F_X:
$$F_X(x) = \mathbb{P}(X \leqslant x)$$
- Kako izgledaju grafici ovih funkcija?

Primer diskretne raspodele

Bacamo nehomogen novčić sve dok ne dobijemo glavu. Glava se javlja sa učestanošću 0.4. Neka je X ukupan broj bacanja. Koja je raspodela promenljive X?

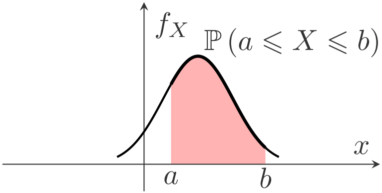
- Očigledno ima smisla isključivo za cele brojeve n.
- $X = n$ ukoliko smo $n - 1$ put dobili pismo i n-ti put dobili glavu.
- $p(n) = 0.6^{n-1} \cdot 0.4$, gde $n \in \mathbb{N}$.
- X ima **geometrijsku raspodelu** sa parametrom 0.4: $X \sim \mathcal{G}(0.4)$.

Gustina verovatnoće

- Ukoliko je skup dopustivih vrednosti *neprekidan* ("unija nekih intervala na $\mathbb{R}$"), i dalje koristimo kumulativnu funkciju raspodele F_X, pa definišemo **funkciju gustine verovatnoće**:

$$f_X = \frac{\mathrm{d}F_X}{\mathrm{d}x}$$

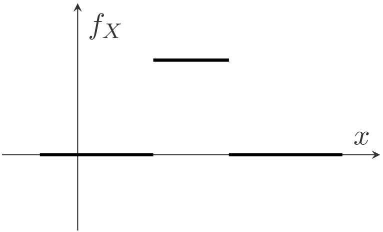
- Značenje: $f_X(x)\,\delta x = \mathbb{P}(x \leqslant X \leqslant x + \delta x)$ kada $\delta x \to 0$.
- Stoga je $\mathbb{P}(a \leqslant X \leqslant b) = \int_a^b f_X(x)\,\mathrm{d}x$.
- Koja su očekivana svojstva gustine?



Uniformna raspodela

- X ima **uniformnu raspodelu** sa parametrima a i b ($a < b$), $X \sim \mathcal{U}(a, b)$, ukoliko:

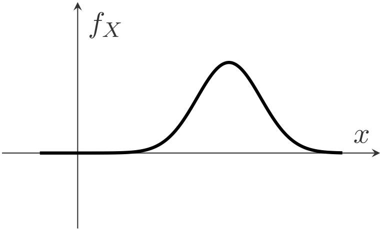
$$f_X(x) = \begin{cases} \frac{1}{b-a}, & a \leqslant x \leqslant b \\ 0, & \text{u suprotnom} \end{cases}$$



Normalna raspodela

- X ima **normalnu raspodelu** sa parametrima μ i σ^2, $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$, ukoliko:

$$f_X\left(x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$



Matematičko očekivanje

- Želimo da vidimo koja je *očekivana* vrednost – ona *nije najverovatnija*, već je "težinska aritmetička sredina". To je **matematičko očekivanje** $\mathbb{E}$.
- Za diskretnu slučajnu promenljivu X koja uzima vrednosti x_1, $x_2, \ldots, x_n, \ldots$ sa verovatnoćama $p_1, p_2, \ldots, p_n, \ldots$ je:
$$\mathbb{E}\left(X\right) = \sum_{n} p_n x_n = \sum_{v} v p\left(v\right)$$
- Za homogenu kockicu očekujemo 3.5, za nehomogenu iz onog primera je $\mathbb{E}\left(X\right) \approx 3.64$.
- U slučaju neprekidne slučajne promenljive:
$$\mathbb{E}\left(X\right) = \int_{-\infty}^{+\infty} x f_X\left(x\right) \mathrm{d}x$$

Matematičko očekivanje (*cont*)

Linearnost očekivanja

Za slučajne promenljive X i Y važi:

$$\mathbb{E}\left(\alpha X + \beta Y\right) = \alpha \mathbb{E}\left(X\right) + \beta \mathbb{E}\left(Y\right)$$

Dokaz. *(samo za diskretnu promenljivu)*

$$\begin{aligned}
\mathbb{E}\left(\alpha X + \beta Y\right) &= \sum_i \sum_j \left[\mathbb{P}\left(X = x_i \wedge Y = y_j\right) \cdot \left(\alpha x_i + \beta y_j\right)\right] \\
&= \alpha \sum_i x_i \sum_j \mathbb{P}\left(X = x_i \wedge Y = y_j\right) \\
&+ \beta \sum_j y_j \sum_i \mathbb{P}\left(X = x_i \wedge Y = y_j\right) \\
&= \alpha \sum_i x_i \mathbb{P}\left(X = x_i\right) + \beta \sum_j y_j \mathbb{P}\left(Y = y_j\right) \\
&= \alpha \mathbb{E}\left(X\right) + \beta \mathbb{E}\left(Y\right)
\end{aligned}$$

□

Odnosi između slučajnih promenljivih

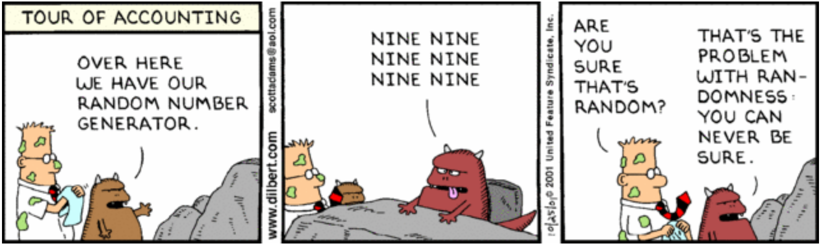
- Ništa nam ne brani da posmatramo i **slučajne vektore**: svaka koordinata je zasebna slučajna promenljiva, (X, Y, Z).
- Međutim, po analogiji sa uslovnom verovatnoćom, možda među njima postoji i neka zavisnost.
- Definišemo da su dve diskretne slučajne promenljive X i Y **nezavisne** ukoliko za sve x i y:
$$\mathbb{P}(X = x \wedge Y = y) = \mathbb{P}(X = x)\,\mathbb{P}(Y = y)$$
- Za neprekidne je malo komplikovanije...
- Može da se pokaže da za nezavisne slučajne promenljive X i Y važi $\mathbb{E}(XY) = \mathbb{E}(X)\,\mathbb{E}(Y)$.
- Veoma često pretpostavljamo nezavisnost promenljivih!

Uslovna raspodela

- Ako su X i Y slučajne promenljive, tada se nekada piše $\mathbb{P}(X \mid Y)$ za sve raspodele X koje se dobijaju pri svim mogućim fiksiranim vrednostima za Y.
- Jedna konkretna raspodela se predstavlja sa $\mathbb{P}(X \mid Y = y)$.
- Opštije: $\mathbb{P}(X \mid Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n)$.
- Značenje: *koja je raspodela X ukoliko su izmerene vrednosti promenljivih $Y_1, Y_2, \ldots, Y_n$ redom $y_1, y_2, \ldots, y_n$?*
- Važe analogne verzije formule totalne verovatnoće i Bajesove formule.

Čemu sve ovo?

- *Ali ovo je bilo predavanje iz matematike, ne informatike! Buhuhu!*
- Iako smo se preleteli preko raznih tema, ovo je tek *najosnovnije* o verovatnoći, odnosno ono što nam je neophodno za naredna dva predavanja.
- U redu je ako vam nije baš sve jasno sada – poslednjih par slajdova služe vam za vežbu!


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
www.dilbert.com scottadams@aol.com
NINE NINE NINE NINE NINE NINE
© 2001 United Feature Syndicate, Inc.
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

Algebra događaja i uslovna verovatnoća

Policija je privela sedmoricu osumnjičenih za pljačku prodavnice nakita. Svedok se seća da su trojica počinila zločin, ali se ne seća tačno o kojoj trojici je reč. Međutim, izvesno je da su sva trojica među osumnjičenima. Svedok zato potpuno proizvoljno, sa jednakim verovatnoćama, identifikuje neku trojicu.

- Koja je verovatnoća da će pogoditi svu trojicu?
- Svedok najpre bira dvojicu osumnjičenih. Koja je verovatnoća da su obojica zaista bili pljačkaši?
- Ako su prva dvojica bila ispravno odabrana, koja je verovatnoća da će i treći biti ispravno odabran?
- Ako je prvi odabran bio pogrešan, koja je verovatnoća da će naredna dva odabrana biti tačna?

Totalna verovatnoća i Bajesova formula

U toku je epidemija mačjeg gripa. Verovatnoća da sam zaražen je 0.001 ($\approx$ svaki hiljaditi je zaražen). Međutim, mogu da se testiram. Bilo da sam zaražen ili nisam, test daje tačan rezultat sa verovatnoćom 0.99.

- Pokazati da je verovatnoća da će test dati rezultat "zaražen" jednaka 0.01098.
- Test je dao rezultat "zaražen". Primenom Bajesove formule pokazati da je mnogo verovatnije da sam u tom slučaju ipak zdrav. Kolika je ta verovatnoća? *(Rezultat bi trebalo da bude ≈ 0.9)*
- Pod pretpostavkom da su rezultati testova nezavisni, kolika je ova verovatnoća ukoliko sam se testirao dva puta, a oba puta sam dobio rezultat "zaražen"?

Diskretna slučajna promenljiva

Date su dve nezavisne diskretne slučajne promenljive X i Y sa raspodelama:

$$X \sim \begin{pmatrix} -1 & 0 & 1 \\ 0.4 & p & 0.3 \end{pmatrix}, \qquad Y \sim \begin{pmatrix} 0 & 1 & 2 \\ 0.1 & 0.5 & q \end{pmatrix}$$

- Odrediti vrednosti p i q.
- Pokazati da $Z = X + Y$ ima raspodelu:
$$Z \sim \begin{pmatrix} -1 & 0 & 1 & 2 & 3 \\ 0.04 & 0.23 & 0.34 & 0.27 & 0.12 \end{pmatrix}$$
- Odrediti raspodelu za $W = XY$.
- Odrediti $\mathbb{E}(X)$, $\mathbb{E}(Y)$, $\mathbb{E}(Z)$, $\mathbb{E}(W)$ na osnovu definicije. Da li je to u skladu sa očekivanim rezultatima?
- Dokazati da za dve nezavisne diskretne slučajne promenljive X i Y važi $\mathbb{E}(XY) = \mathbb{E}(X)\,\mathbb{E}(Y)$.

Probabilističke metode i mašinsko učenje

Markovljevi lanci, PageRank i grupisanje

Petar Veličković

Matematička gimnazija
Nedelja informatike v2.0

16. decembar 2015.

Markovljevi lanci

- **Markovljev lanac** (*Markov chain*) je, neformalno, nasumičan (*stohastičan*) proces, nad diskretnim skupom stanja, bez interne memorije (u toku predavanja ćemo dati preciznu definiciju).
- Teoretske osnove postavio Andrej Markov, 1906. godine.
- Izuzetno jednostavan model (u smislu složenosti stanja koje se mora pamtiti); i dalje primenljiv na mnoge oblasti širom nauke!
- Zahteva poznavanje samo osnovnih koncepata verovatnoće i linearne algebre (uslovna verovatnoća, množenje matrica).

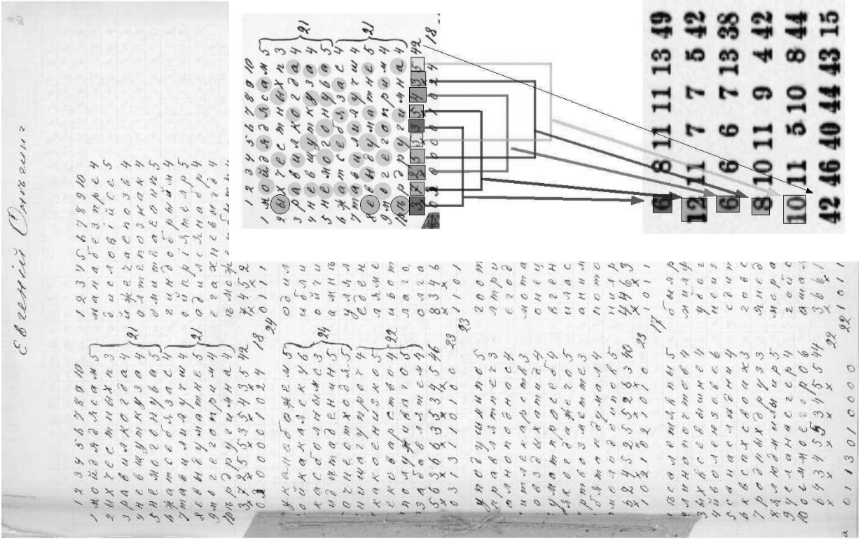
Primene Markovljevih lanaca

- Markovljevi lanci imaju mnogo više (direktnih ili indirektnih) primena nego što može stati na jedan slajd.
- Ovde ćemo navesti samo najvažnijih pet.

Pet najvažnijih primena

1. Andrej Markov (1913)—**analiza rasprostranjenosti samoglasnika i suglasnika u ruskom jeziku** (korišćen "Evgenije Onjegin" od Puškina);
2. Klod Šenon (1948)—**modeliranje Engleskog jezika, u svrhu analize njegove entropije** (radi konstrukcije optimalnih kodiranja $\sim$ Morzeova azbuka; prvo pominjanje reči bit!);
3. Alan Šer (1965)—**modeliranje distribuiranog računarskog sistema** (pokazano da sa daleko jednostavnijim modelom može da se vrlo precizno modelira sistem);
4. Leonard Baum (1966)—**skriveni Markovljevi modeli** (prvobitno korišćeni za prepoznavanje govora, sada primenljivi preko čitavog spektra mašinskog učenja);
5. Sergej Brin i Leri Pejdž (1996)—**PageRank** (prvi i najpoznatiji algoritam (i dan danas) korišćen za pretraživanje na Guglu).

Originalna primena (Markov, 1913)



Markovljevo stvojstvo

- Neka je S konačan skup stanja, a $\{X_n\}_{n\geq 0}$ niz slučajnih promenljivih koje uzimaju vrednosti iz S.
- Ovaj niz zadovoljava **Markovljevo svojstvo** ukoliko je *bezmemorijski*: ukoliko naredna vrednost u nizu zavisi samo od trenutne; tj. za sve $n \geq 0$:

$$\begin{aligned}&\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n, \ldots, X_0 = x_0)\\ =\ &\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)\end{aligned}$$

  On se tada naziva *Markovljevim lancem*.
- $X_t = x_t$ znači da je lanac *u momentu* t *u stanju* x_t $(\in S)$.

Vremenska homogenost

- Česta pretpostavka je **vremenska homogenost** lanaca—da se verovatnoće prelaza ne menjaju sa vremenom; tj. za sve $n \geq 0$:

$$\mathbb{P}(X_{n+1} = b | X_n = a) = \mathbb{P}(X_1 = b | X_0 = a)$$

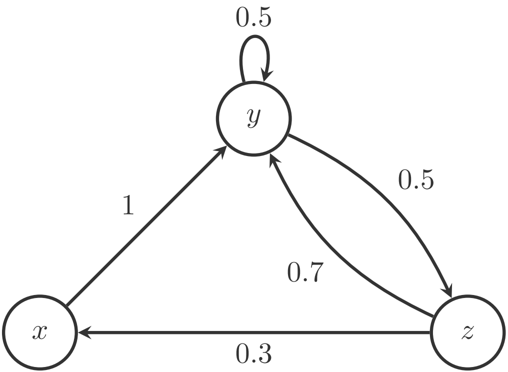
- Vremenska homogenost nam dozvoljava da predstavimo Markovljeve lance sa konačnim skupom stanja S samo jednom matricom, $\mathbf{P}$:

$$\mathbf{P}_{ij} = \mathbb{P}(X_1 = j | X_0 = i)$$

  za svako $i, j \in S$. Važi $\sum_{x \in S} \mathbf{P}_{ix} = 1$ za svako $i \in S$.

- Takođe možemo takve lance predstaviti kao *probabilističke konačne automate* (slično kao u prošlogodišnjem predavanju).

Primer Markovljevog lanca



$$S = \{x, y, z\}$$

$$\mathbf{P} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0.0 \end{bmatrix}$$

Matrica prelaza u n koraka

- Sada nas zanima kolika je verovatnoća da, nakon napravljenih n koraka, pređemo iz stanja i u stanje j. Ovo definiše novu matricu $\mathbf{P}^{(n)}$, tako da

$$\mathbf{P}_{ij}^{(n)} = \mathbb{P}(X_n = j | X_0 = i)$$

- Kako izgleda $\mathbf{P}^{(1)}$? A $\mathbf{P}^{(0)}$?

Matrica prelaza u n koraka

- Sada nas zanima kolika je verovatnoća da, nakon napravljenih n koraka, pređemo iz stanja i u stanje j. Ovo definiše novu matricu $\mathbf{P}^{(n)}$, tako da

$$\mathbf{P}_{ij}^{(n)} = \mathbb{P}(X_n = j | X_0 = i)$$

- Kako izgleda $\mathbf{P}^{(1)}$? A $\mathbf{P}^{(0)}$?

- $\mathbf{P}^{(1)} = \mathbf{P}$;
  $\mathbf{P}^{(0)} = \mathbf{I}$.

Čepmen-Kolmogorova teorema

Teorema (Čepmen-Kolmogorov)

Za sva stanja i, j i nenegativne cele brojeve n, m, važi:

$$\mathbf{P}_{ij}^{(n+m)} = \sum_{k \in S} \mathbf{P}_{ik}^{(n)} \mathbf{P}_{kj}^{(m)}$$

Posledica

$$\mathbf{P}^{(n+m)} = \mathbf{P}^{(n)} \mathbf{P}^{(m)} \implies \mathbf{P}^{(n)} = \mathbf{P}^{n}$$

Dakle, matricu prelaza u n koraka dobijemo stepenovanjem početne matrice na n-ti stepen.

Čepmen-Kolmogorova teorema

Dokaz

$$
\begin{aligned}
\mathbf{P}_{ij}^{(n+m)} &= \mathbb{P}(X_{n+m} = j | X_0 = i) \\
&= \sum_{k \in S} \mathbb{P}(X_{n+m} = j, X_n = k | X_0 = i) \\
&= \sum_{k \in S} \mathbb{P}(X_{n+m} = j | X_n = k, X_0 = i)\mathbb{P}(X_n = k | X_0 = i) \\
&= \sum_{k \in S} \mathbb{P}(X_{n+m} = j | X_n = k)\mathbb{P}(X_n = k | X_0 = i) \\
&= \sum_{k \in S} \mathbb{P}_{ik}^{(n)} \mathbf{P}_{kj}^{(m)}
\end{aligned}
$$

□

Komunicirajuće klase

- Ukoliko za neko $n \geq 0$ važi $\mathbf{P}_{ij}^{(n)} > 0$, kažemo da je stanje j *dostupno* iz stanja i; označavamo sa $i \rightsquigarrow j$.
- Ukoliko $i \rightsquigarrow j$ i $j \rightsquigarrow i$, onda kažemo da stanja i i j *komuniciraju*; označavamo sa $i \leftrightsquigarrow j$.
- Lako je primetiti da je relacija $\leftrightsquigarrow$:
  - *refleksivna* ($i \leftrightsquigarrow i$)
  - *simetrična* ($i \leftrightsquigarrow j \implies j \leftrightsquigarrow i$)
  - *tranzitivna* ($i \leftrightsquigarrow j \wedge j \leftrightsquigarrow k \implies i \leftrightsquigarrow k$)
- Dakle, ova relacija je relacija *ekvivalencije*, i samim tim deli skup stanja u *klase ekvivalencije*, koje ćemo nazvati *komunicirajućim klasama*.
- Ukoliko Markovljev lanac ima samo jednu komunicirajuću klasu, kažemo da je *nesvodljiv* (*irreducible*).

Periodičnost

- Razmatrajmo sve momente kada, krenuvši iz stanja i, možemo da se vratimo u stanje i (tj. sve momente t kada važi $\mathbf{P}_{ii}^{(t)} > 0$).
- Ukoliko je NZD (najveći zajednički delilac) svih ovih momenata jednak 1, onda kažemo da je stanje i *aperiodično*.
- U suprotnom, ukoliko je NZD jednak $d > 1$, kažemo da je stanje i *periodično sa periodom* d.
- Može se pokazati da je periodičnost *svojstvo komunicirajućih klasa*, tj. $i \leftrightsquigarrow j \implies d_i = d_j$.

Stacionarna raspodela

- Ukoliko označimo sa $\vec{\lambda}$ vektor koji predstavlja verovatnoće da se nalazimo u svakom od stanja u nekom trenutku (mora da važi $\sum_{k \in S} \lambda_k = 1$), onda nakon jednog koraka te verovatnoće postaju
$$\vec{\lambda}' = \vec{\lambda}\mathbf{P}$$
- *Stacionarna raspodela* za neki Markovljev lanac predstavlja vektor verovatnoća $\vec{\pi}$ koji se ne menja nakon jednog koraka, tj.
$$\vec{\pi} = \vec{\pi}\mathbf{P}$$
- Stacionarna raspodela, dakle, predstavlja *prosečnu proporciju vremena* koju će lanac provesti u svakom od stanja.

Svojstva stacionarne raspodele

- Ukoliko Markovljev lanac ima stacionarnu raspodelu, onda će raspodela stanja uvek težiti stacionarnoj nakon određenog broja koraka:

$$\forall \vec{\lambda}. \lim_{n \to +\infty} \vec{\lambda} \mathbf{P}^n = \vec{\pi}$$

- Iz ovoga se može direktno izvesti izgled matrice prelaza u n koraka, za dovoljno veliko n:

$$\lim_{n \to +\infty} \mathbf{P}_{ij}^n = \pi_j$$

- Nemaju svi Markovljevi lanci stacionarnu raspodelu; za naše potrebe, smatraćemo da lanac mora biti nesvodljiv i aperiodičan da bi $\vec{\pi}$ postojao (pravi uslov je malo strožiji).

Primer (Markov, 1913)

- Markov je proučavao rasprostranjenost samoglasnika i suglasnika u "Evgeniju Onjeginu". Direktnim merenjem je utvrdio sledeću matricu prelaza (skup stanja je $\{samog, sugl\}$):

$$\mathbf{P} = \begin{bmatrix} 0.128 & 0.872 \\ 0.663 & 0.337 \end{bmatrix}$$

- Prvih nekoliko stepena $\mathbf{P}$:

$$\mathbf{P}^2 = \begin{bmatrix} 0.595 & 0.405 \\ 0.308 & 0.692 \end{bmatrix} \quad \mathbf{P}^3 = \begin{bmatrix} 0.345 & 0.655 \\ 0.498 & 0.502 \end{bmatrix} \quad \mathbf{P}^4 = \begin{bmatrix} 0.478 & 0.522 \\ 0.397 & 0.603 \end{bmatrix}$$

Primer (Markov, 1913)

- Moguće je utvrditi da važi

$$\lim_{n \to +\infty} \mathbf{P}^n = \begin{bmatrix} 0.432 & 0.568 \\ 0.432 & 0.568 \end{bmatrix}$$

- Odavde možemo direktno izvesti stacionarnu raspodelu, tj.

$$\vec{\pi} = \begin{bmatrix} 0.432 & 0.568 \end{bmatrix}$$

- *Za vežbu*: Direktno proveriti da važi $\vec{\pi}\mathbf{P} = \vec{\pi}$.

Pretraživanje interneta

- **Problem:** *dobaviti* stranice sa interneta koje "najbolje" zadovoljavaju neki upit korisnika, i *rangirati* ih po relevantnosti.
- Konvencionalni pretraživači tokom 90-tih godina prebrojavaju koliko puta se termini u upitu pojavljuju na stranicama.
- U januaru 1996. Leri Pejdž i Sergej Brin, doktorski studenti na Stenfordu, postavljaju *PageRank*, prvi model relevantnosti koji uzima u obzir *veze* između stranica.
- Ovaj model je bio centralni u jednom, sada svetski poznatom, internet pretraživaču…

Google, kad je još imao gotivan logo

Pretpostavke

- Internet možemo predstaviti kao **usmeren graf**, koristeći *stranice* kao čvorove i *hiperveze* kao ivice.

- U sklopu ovog modela pravimo neke pretpostavke:
    1. Hiperveza je pokazatelj kvaliteta; $d_1 \to d_2$ znači da vlasnik sajta d_1 smatra da je sajt d_2 *visokog kvaliteta i relevantan*.
    2. Tekst oko hiperveze (*anchor text*) potpuno opisuje stranicu d_2.
    3. Prosečni korisnici interneta se ponašaju kao *nasumični šetači* po ovom grafu.

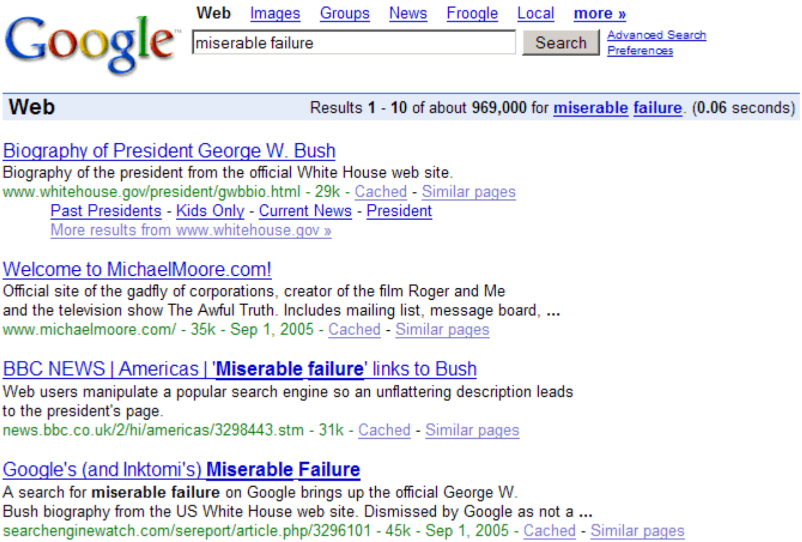
- Mnoge od ovih pretpostavki su verovatno bile sasvim validne u 1998; danas *Google* koristi PageRank kao samo jedan od mnogobrojnih faktora pri rangiranju stranica.

Kratko skretanje: *Google bombs*

- Kada je *Google* pridavao pun značaj tekstu oko hiperveza i strukturi ovog usmerenog grafa, manipulacije rezultata su bile jednostavne; samo je bilo potrebno napraviti mnogo hiperveza sa željenim okolnim tekstom.
- Ove manipulacije su popularno nazivane *Google bombama*. Popularni primeri (od kojih su neki i danas aktivni):

"miserable failure"	$\longrightarrow$	Džordž Buš, Majkl Mur;
"dumb motherf… "	$\longrightarrow$	Džordž Buš;
"evil empire"	$\longrightarrow$	*Microsoft*;
"dangerous cult"	$\longrightarrow$	Sajentološka crkva;
"french military victories"	$\longrightarrow$	"Did you mean: French military defeats?".

Miserable failure

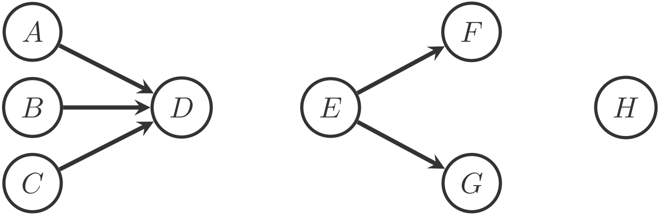


Nasumični surfer

- *PageRank* se zasniva na pretpostavci da se korisnici ponašaju kao *nasumični surferi*; nakon što su boravili neko vreme na nekoj stranici, odabraće nasumičnu hipervezu i preći na sledeću stranicu.
- Hiperveze se biraju *uniformno*; ako na trenutnoj stranici imamo n hiperveza, verovatnoća da će korisnik odabrati neku (konkretnu) hipervezu je $\frac{1}{n}$.
- Ovime smo efektivno izgradili *Markovljev lanac*! Ako pronađemo njegovu stacionarnu raspodelu, dobijamo prosečnu proporciju vremena koju će surfer provesti na svakom sajtu, što je odlična mera relevantnosti—upravo ova raspodela predstavlja *PageRank* neke stranice.
- Da li neko vidi problem sa ovim modelom?

Ćorsokaci

- Internet je pun *ćorsokaka*, tj. stranica bez hiperveza.
- Njihovo postojanje uglavnom izaziva da stacionarna raspodela *ne bude jedinstveno definisana* (npr. da zavisi od toga iz kog čvora surfer započinje). Primer:



Teleportovanje

- Problem ćorsokaka rešavamo *teleportovanjem*; u svakom momentu, sa nekom (malom) verovatnoćom, α, skačemo na **bilo koju** stranicu, umesto praćenja hiperveze.
- Specijalno, ukoliko smo u ćorsokaku, skačemo na bilo koju stranicu sa uniformnom verovatnoćom.
- Formalnije, matrica prelaza za naš Markovljev lanac je oblika:

$$\mathbf{P}_{ij} = \begin{cases} \frac{1}{n} & \ell_i = 0 \\ \alpha\frac{1}{n} & \ell_i > 0 \wedge i \not\to j \\ \alpha\frac{1}{n} + (1-\alpha)\frac{1}{\ell_i} & \ell_i > 0 \wedge i \to j \end{cases}$$

gde sa ℓ_i označavamo broj hiperveza iz stranice i, a sa n ukupan broj stranica.

Metoda stepenovanja

- **Podsetnik:**
  Ukoliko Markovljev lanac ima stacionarnu raspodelu, onda će raspodela stanja uvek težiti stacionarnoj nakon određenog broja koraka:
  $$\forall \vec{\lambda}. \lim_{n \to +\infty} \vec{\lambda} \mathbf{P}^n = \vec{\pi}$$

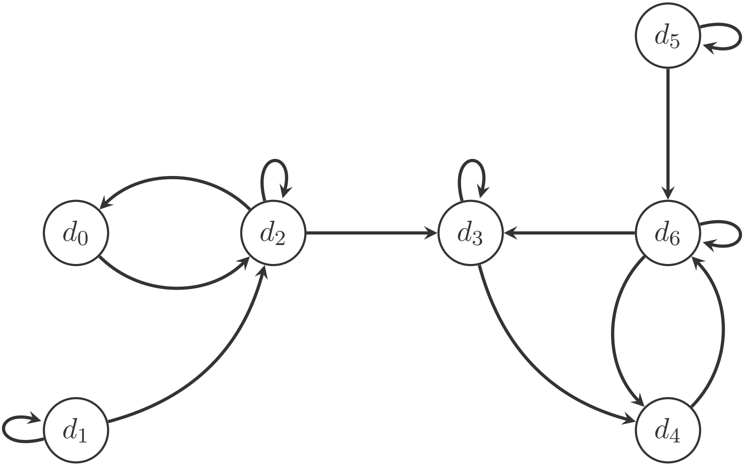
- Ovo nam daje direktnu iterativnu metodu za računanje *PageRank*-a—**metodu stepenovanja** (*power method*).

- Ova metoda se vrlo lako paralelizuje, i stoga je pogodna za računanje na distribuiranim sistemima (kakve ima *Google*).

Implementacija

- Metoda stepenovanja:
  1. Početi sa nasumičnom raspodelom, $\vec{\lambda}_0$, $t \leftarrow 0$;
  2. $\vec{\lambda}_{t+1} \leftarrow \vec{\lambda}_t \mathbf{P}$;
  3. Ako je $|\vec{\lambda}_{t+1} - \vec{\lambda}_t| < \varepsilon$, kraj.
  4. U suprotnom, $t \leftarrow t + 1$, nazad na korak 2.
- Naivna implementacija ima vremensku složenost $O(t \cdot n^2)$ i memorijsku složenost $O(n^2)$; međutim, ukoliko napravimo pametne optimizacije, moguće je ove složenosti spustiti na $O(t \cdot (n + \ell))$ (vremenska) i $O(n + \ell)$ (memorijska), gde je ℓ broj hiperveza.
- Implementaciju ove metode (u C++) možete naći na: `https://github.com/PetarV-/Algorithms/blob/master/Mathematical%20Algorithms/PageRank.cpp`

Primer grafa interneta



Matrica susednosti, naivna matrica prelaza

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}_{naive} = \begin{bmatrix} 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.33 & 0.00 & 0.33 & 0.33 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.50 \\ 0.00 & 0.00 & 0.00 & 0.33 & 0.33 & 0.00 & 0.33 \end{bmatrix}$$

Matrica prelaza sa teleportovanjem

$$\alpha = 0.1$$

$$\mathbf{P} = \begin{bmatrix} 0.014 & 0.014 & 0.914 & 0.014 & 0.014 & 0.014 & 0.014 \\ 0.014 & 0.464 & 0.464 & 0.014 & 0.014 & 0.014 & 0.014 \\ 0.311 & 0.014 & 0.311 & 0.311 & 0.014 & 0.014 & 0.014 \\ 0.014 & 0.014 & 0.014 & 0.464 & 0.464 & 0.014 & 0.014 \\ 0.014 & 0.014 & 0.014 & 0.014 & 0.014 & 0.014 & 0.914 \\ 0.014 & 0.014 & 0.014 & 0.014 & 0.014 & 0.464 & 0.464 \\ 0.014 & 0.014 & 0.014 & 0.311 & 0.311 & 0.014 & 0.311 \end{bmatrix}$$

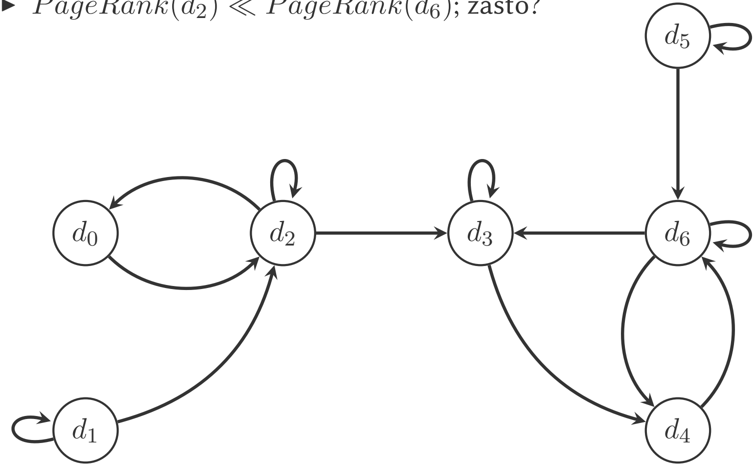
Rezultati

```
Iteration 0:    {0.142857, 0.142857, 0.142857, 0.142857, 0.142857, 0.142857, 0.142857}
Iteration 1:    {0.057143, 0.078571, 0.250000, 0.164286, 0.121429, 0.078571, 0.250000}
Iteration 2:    {0.089286, 0.049643, 0.176071, 0.238214, 0.163214, 0.049643, 0.233929}
Iteration 3:    {0.067107, 0.036625, 0.169804, 0.244482, 0.191661, 0.036625, 0.253696}
Iteration 4:    {0.065227, 0.030767, 0.142104, 0.251353, 0.200412, 0.030767, 0.279371}
Iteration 5:    {0.056917, 0.028131, 0.129466, 0.253837, 0.211206, 0.028131, 0.292312}
Iteration 6:    {0.053126, 0.026945, 0.117010, 0.255046, 0.216206, 0.026945, 0.304723}
Iteration 7:    {0.049389, 0.026411, 0.109327, 0.255576, 0.220473, 0.026411, 0.312413}
Iteration 8:    {0.047084, 0.026171, 0.103418, 0.255817, 0.223019, 0.026171, 0.318321}
Iteration 9:    {0.045311, 0.026062, 0.099463, 0.255925, 0.224900, 0.026062, 0.322276}
Iteration 10:   {0.044125, 0.026014, 0.096633, 0.255974, 0.226135, 0.026014, 0.325106}
Iteration 11:   {0.043276, 0.025992, 0.094694, 0.255996, 0.227006, 0.025992, 0.327045}
Iteration 12:   {0.042694, 0.025982, 0.093338, 0.256005, 0.227597, 0.025982, 0.328401}
Iteration 13:   {0.042287, 0.025978, 0.092404, 0.256010, 0.228008, 0.025978, 0.329335}
Iteration 14:   {0.042007, 0.025976, 0.091755, 0.256012, 0.228291, 0.025976, 0.329984}
Program ended with exit code: 0
```

$$\vec{\pi} \approx \begin{bmatrix} 0.04 & 0.03 & 0.09 & 0.25 & 0.23 & 0.03 & 0.33 \end{bmatrix}$$

Diskusija

- $\vec{\pi} \approx \begin{bmatrix} 0.04 & 0.03 & 0.09 & 0.25 & 0.23 & 0.03 & 0.33 \end{bmatrix}$
- $PageRank(d_2) \ll PageRank(d_6)$; zašto?

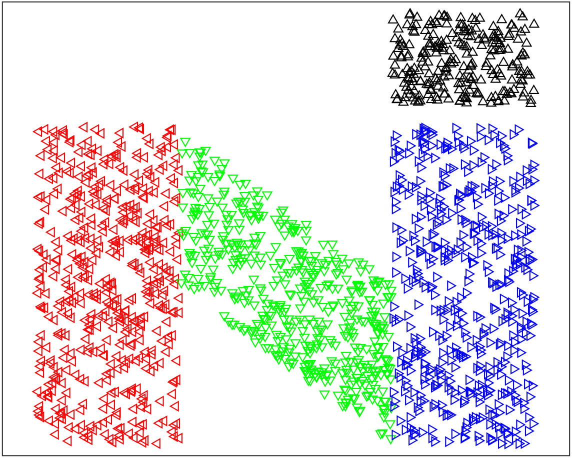


Grupisanje

- Problem **grupisanja** (*clustering*) predstavlja potrebu da se određen skup podataka podeli u podskupove, tako da su elementi unutar jednog podskupa međusobno *"sličniji"* u odnosu na elemente unutar drugih.
- Ovo već predstavlja neku vrstu *mašinskog učenja*! Mašina treba da, samo na osnovu ulaznih elemenata, "odluči" koji element pripada kom podskupu—kao i, ponekad, koliko podskupova će biti u konačnom rešenju.
- *Širok* spektar primena:
  - Bioinformatika (prepoznavanje *genetskih familija*);
  - Medicina (prepoznavanje različitih vrsta *tkiva* i *krvi* na PET slici);
  - Socijalne mreže (prepoznavanje ljudskih *zajednica*);
  - Kriminalistika (prepoznavanje *hot-spotova* za neke zločine), ...

Primer grupisanja

Grupisanje je najlakše zamisliti u 2D prostoru, npr:



UPGMA i MCL

- Na prethodnoj Nedelji informatike (u okviru predavanja o *Bioinformatičkim algoritmima*) rađen je *UPGMA* algoritam, koji izgrađuje grupisanje tako što je u početku svaki element u svom skupu, i u svakom koraku spaja najsličnija dva skupa u jedan.
- Algoritam koji ćemo detaljnije pokriti sada, **Markovljevo grupisanje** (*Markov clustering; MCL*), koristi obrnut pristup: u početku su svi elementi u jednom skupu, pa se skupovi potencijalno naknadno "parčaju".

MCL

- *MCL* uzima kao ulaz *težinski graf*, a kao izlaz daje *grupisanje njegovih čvorova.*
- Težina ivice ističe "povezanost" ili "sličnost" dva čvora koja povezuje. Ukoliko graf nije težinski, smatramo da je težina svake ivice 1.
- Ovaj graf možemo posmatrati kroz njegovu matricu povezanosti, $\mathbf{M}$. Dodatno ćemo *normalizovati ovu matricu* (tako da se elementi u svakom redu sumiraju u 1):

$$\mathbf{M}'_{ij} = \frac{\mathbf{M}_{ij}}{\sum_k \mathbf{M}_{ik}}$$

Ovim smo pretvorili ulazni graf u *Markovljev lanac*!

Pretpostavke MCL-a

1. Unutar jedne grupe postoji “daleko više” puteva nego između grupa. Ekvivalentno: nasumični putnik će verovatnije ostati unutar grupe u kojoj se trenutno nalazi.

2. Elementi unutar jedne grupe su “daleko jače” povezani od elemenata iz različitih grupa.

3. Previše slabe veze nemaju samerljiv uticaj na grupisanje.

Pretpostavke MCL-a

1. Unutar jedne grupe postoji “daleko više” puteva nego između grupa. Ekvivalentno: nasumični putnik će verovatnije ostati unutar grupe u kojoj se trenutno nalazi.

   $\Longrightarrow$ U svakom koraku algoritma računamo koliko jako su čvorovi povezani putevima dužine e (*expansion step*)—ovo je ekvivalentno računanju $\mathbf{M}^e$!

2. Elementi unutar jedne grupe su “daleko jače” povezani od elemenata iz različitih grupa.

   $\Longrightarrow$ Zatim, primenimo operaciju koja će ojačati jake ivice a oslabiti slabe ivice: stepenujemo težinu svake ivice u matrici sa r i ponovo normalizujemo (*inflation step*).

3. Previše slabe veze nemaju samerljiv uticaj na grupisanje.

   $\Longrightarrow$ Tokom izvršenja algoritma možemo na nulu postaviti ivice čija jačina postane manja od ε.

MCL

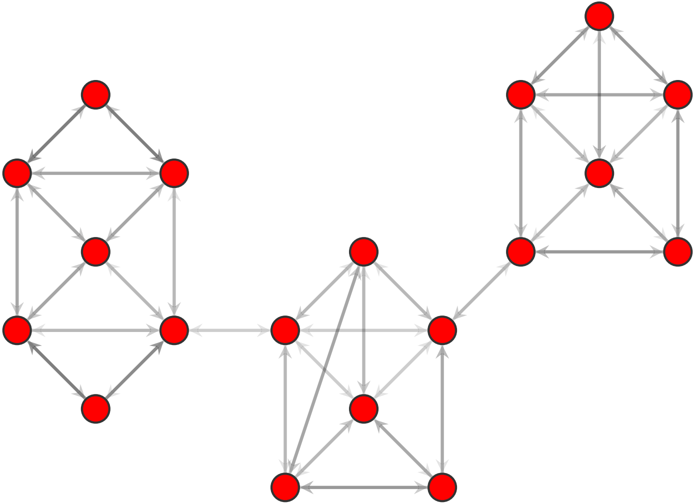
MCL algoritam:

1. $\mathbf{M}_0 \leftarrow \mathbf{M}', t \leftarrow 0$;
2. $\mathbf{M}_{t+1} \leftarrow (\mathbf{M}_t)^e$ (*expansion step*);
3. Za sve i, j:
   3.1 $(\mathbf{M}_{t+1})_{ij} \leftarrow \frac{(\mathbf{M}_{t+1})_{ij}^r}{\sum_k (\mathbf{M}_{t+1})_{ik}^r}$ (*inflation step*);
   3.2 Ako je $(\mathbf{M}_{t+1})_{ij} < \varepsilon$, $(\mathbf{M}_{t+1})_{ij} \leftarrow 0$;
4. $t \leftarrow t + 1$;
5. Ukoliko ima promena u vezama, nazad na korak 2.
   U suprotnom, ispisati svaku povezanu komponentu $\mathbf{M}_t$ kao jednu grupu.

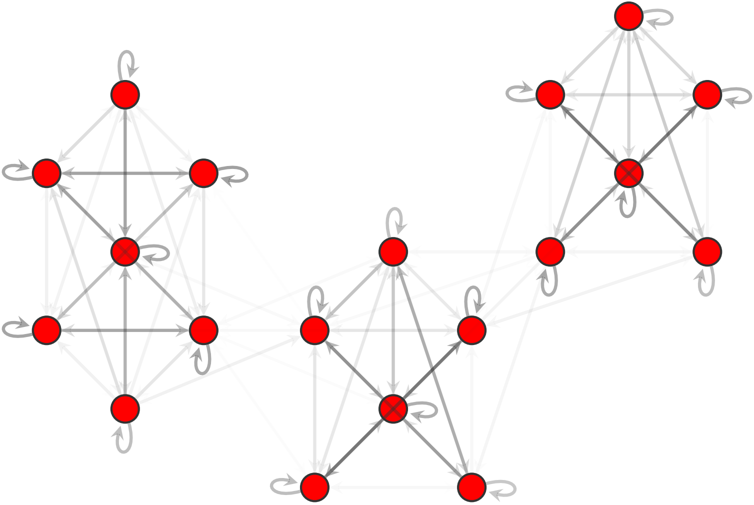
Implementacioni detalji

- Uglavnom algoritam brzo konvergira (do 100 koraka), i matrice nakon nekoliko koraka postaju retke.
- Često se koriste parametri $e = r = 2$, međutim u praksi se algoritam pokreće više puta sa raznim vrednostima parametara, nakon čega se odabira najpogodniji izlaz.
- Naivna implementacija ima složenost $O(n^3 \log e)$ za ekspanziju i $O(n^2 \log r)$ za inflaciju—međutim, ukoliko su matrice retke, moguće je drastično ubrzati ove algoritme.
- Implementaciju ove metode (u C++) možete naći na: `https://github.com/PetarV-/Machine-Learning/blob/master/Clustering/Markov%20Clustering.cpp`

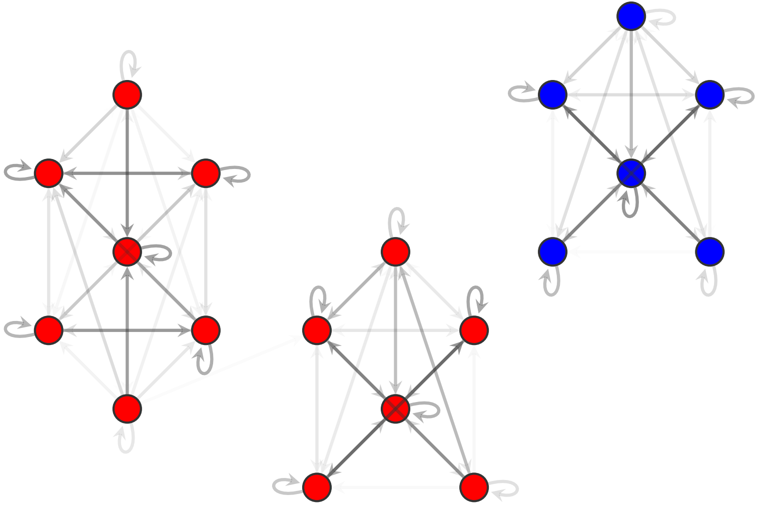
Primer ulaznog grafa



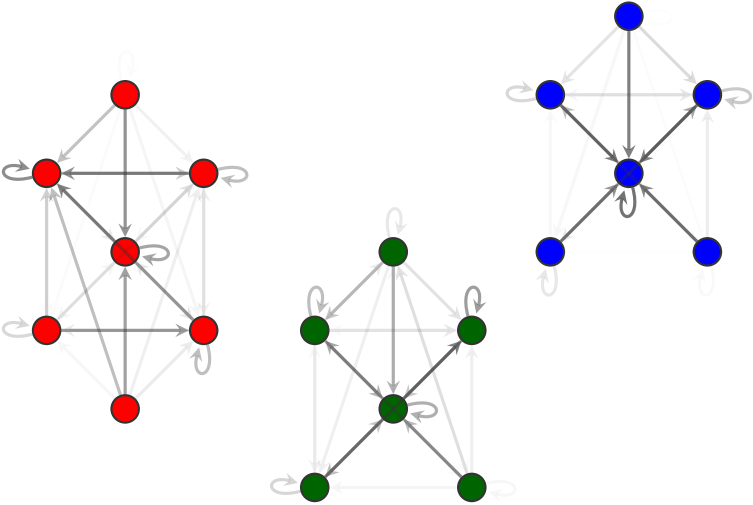
Primer ulaznog grafa: iteracija 1



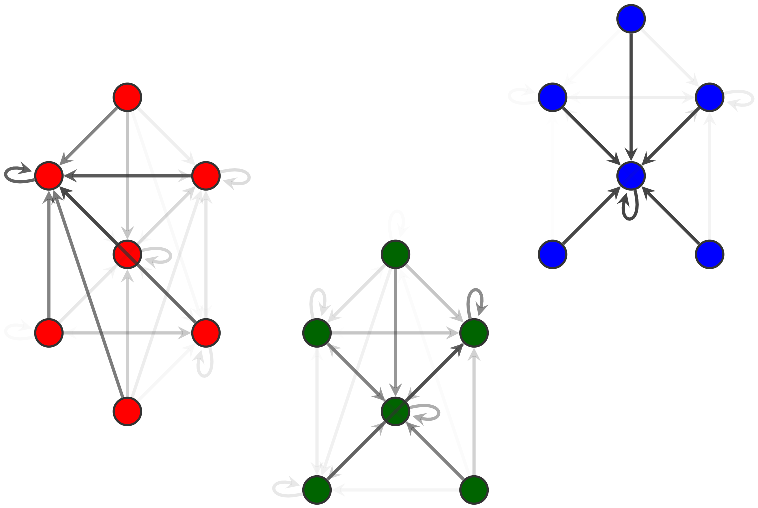
Primer ulaznog grafa: iteracija 2



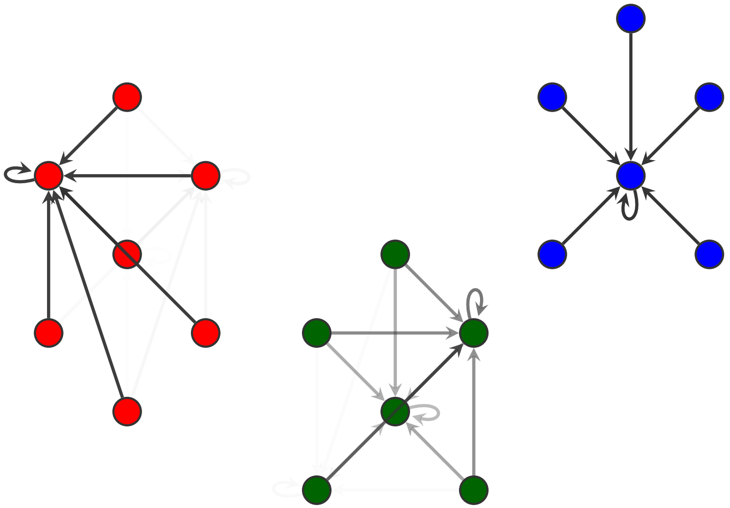
Primer ulaznog grafa: iteracija 3



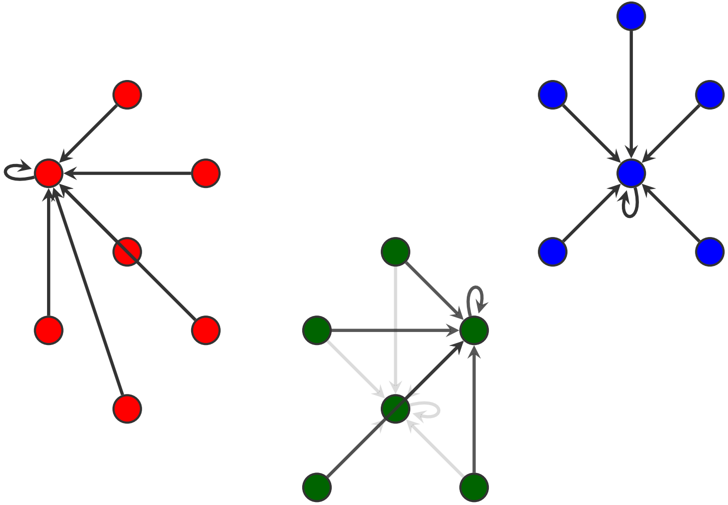
Primer ulaznog grafa: iteracija 4



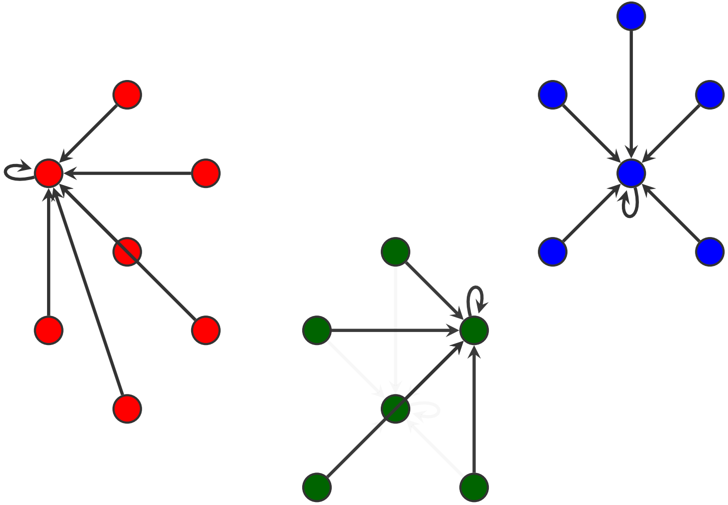
Primer ulaznog grafa: iteracija 5



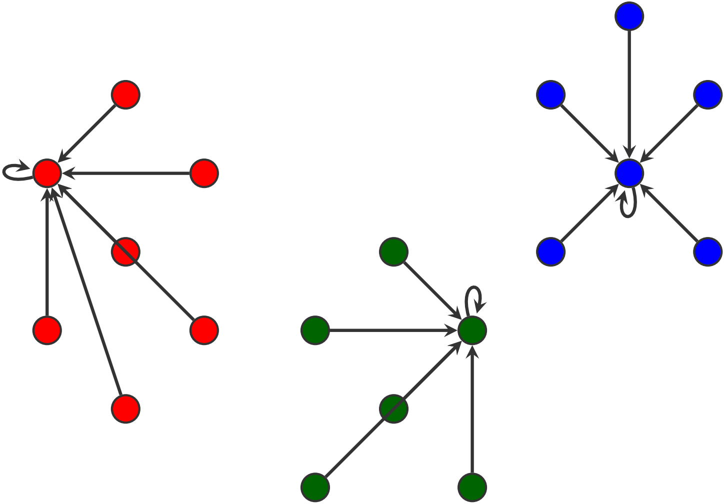
Primer ulaznog grafa: iteracija 6



Primer ulaznog grafa: iteracija 7



Primer ulaznog grafa: iteracija 8



Probabilističke metode i mašinsko učenje

Rezonovanje u vremenu, skriveni Markovljevi modeli

Andrej Ivašković

Matematička gimnazija
Nedelja informatike v2.0

16. decembar 2015.

Koncept vremena u veštačkoj inteligenciji

- U jednostavnijim poglavljima mašinskog učenja (npr. neuralne mreže), sistem koji se proučava se smatra *nepromenljivim*.
- Neretko ima smisla posmatrati *procese* koji se menjaju u *vremenu*, prolaze kroz neka "nevidljiva" **stanja** i "vidljive" **obzervacije**.
- Razmatraćemo **skrivene Markovljeve modele**. Postoje i *Kalmanovo filtriranje*, *dinamičke Bajesove mreže.*
- Čest primer: *prepoznavanje ljudskog govora.*

Napomene u vezi sa oznakama

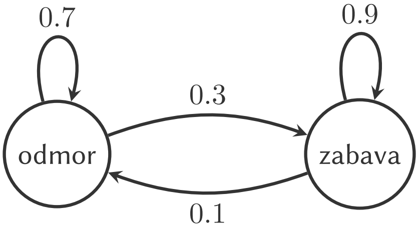
- Oznake koje ćemo nadalje koristiti će biti malo nezgrapne, ali neophodne radi jednostavnosti i kompaktnosti.
- $A_{1:n}$ je skraćeni zapis za $A_1, A_2, \ldots, A_n$.
- $\mathbb{P}(\omega \mid e_k)$ je skraćeni zapis za $\mathbb{P}(\omega \mid E_k = e_k)$.
- Ako do sada niste videli, $\arg\max_{s \in V} f(s)$ je vrednost s iz nekog dozvoljenog (uglavnom podrazumevanog) skupa vrednosti V za koju je vrednost f najveća moguća tj.
$$\max_{x \in V} f(x) = f\left(\arg\max_{s \in V} f(s)\right)$$

Stacionarni i Markovljevi procesi

- Pretpostavljamo diskretno vreme, odnosno "parčiće" vremena $1, 2, \ldots, t$, a stanja su data vrednostima diskretnih slučajnih promenljivih $X_1, X_2, \ldots, X_t$.
- Pretpostavljamo da je proces **stacionaran**: pravila promena stanja se ne menjaju u vremenu. Porediti sa **statičnim** procesom.
- Specijalno, imamo i **Markovljevo svojstvo**: prelaz u naredno stanje zavisi isključivo od trenutnog. Drugačije zapisano:
$$\mathbb{P}\left(X_{t+1} \mid X_{0:t}\right) = \mathbb{P}\left(X_{t+1} \mid X_t\right)$$
- U suštini, posmatramo Markovljeve lance iz malo drugačije perspektive.

Primer Markovljevog procesa

Zli gospodar Artur se ili odmara ili zabavlja, a potencijalno menja režim rada na svakih sat vremena. Ako se odmara, sa verovatnoćom 0.7 nastavlja sa odmorom. Ako se zabavlja, sa verovatnoćom 0.9 nastavlja sa zabavljanjem. Kada se odmara, stvari eksplodiraju sa verovatnoćom 0.05. Kada se zabavlja, ova verovatnoća je 0.4.



Prelazi i senzori

- Neka promenljive X_t predstavljaju **stanja** (zabava i odmor), a E_t predstavljaju **obzervacije** (ima eksplozije, nema eksplozije). Konkretna stanja i obzervacije su x_t i e_t.
- **Model prelaza**: uslovna raspodela $\mathbb{P}\left(X_{t+1} \mid X_t\right)$.
- **Senzorski model**: uslovna raspodela $\mathbb{P}\left(E_t \mid X_t\right)$.

$$
\begin{array}{ccccccccc}
\cdots & \rightarrow & X_{t-1} & \rightarrow & X_t & \rightarrow & X_{t+1} & \rightarrow & \cdots \\
 & & \downarrow & & \downarrow & & \downarrow & & \\
 & & E_{t-1} & & E_t & & E_{t+1} & &
\end{array}
$$

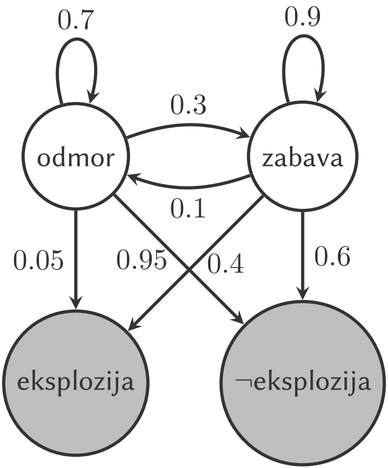
Šta je skriveni Markovljev model?

- Vremenski-zavisan proces u kom je stanje predstavljeno jednom jedinom diskretnom slučajnom promenljivom koja uzima konačan broj vrednosti, a dostupne su obzervacije koje zavise isključivo od aktuelnog stanja.
- Podsetnik: Markovljevi lanci mogu da se predstavljaju $n \times n$ matricama. Ovde ćemo matricu prelaza označiti sa $\mathbf{T}$.
- Alternativna interpretacija: menjaju se stanja u Markovljevom lancu, ali u svakom stanju se na raspolaganje stavlja neki od mogućih izlaza.

Šta nam je potrebno za kompletnu definiciju HMM?

- $\vec{\pi}$: raspodela verovatnoća za početno stanje
- $\mathbf{T}$: matrica prelaza za Markovljev lanac
- $\mathbf{O}$: matrica raspodela verovatnoća za izlaze tj. $\mathbf{O}_{xe} = \mathbb{P}\left(E_0 = e \mid X_0 = x\right)$

Zadatak kao skriveni Markovljev model



Osnovni zadaci

- **Filtriranje**: odrediti $\mathbb{P}\left(X_t \mid e_{1:t}\right)$ tj. raspodelu verovatnoća za aktuelno stanje ako se uzmu u obzir sve obzervacije do sada. Na osnovu ovoga se donose *racionalne* odluke.
- **Predviđanje**: odrediti $\mathbb{P}\left(X_{t+k} \mid e_{1:t}\right)$ za neko $k \in \mathbb{N}$.
- **Učenje**: za uočene izlaze $e_1, e_2, \ldots, e_t$, odrediti parametre skrivenog Markovljevog modela tako da se maksimizuje $\mathbb{P}\left(e_{1:t} \mid \text{parametri HMM}\right)$.

Osnovni zadaci (nastavak)

- **Najverovatnije objašnjenje (dekodiranje)**: za date obzervacije, odrediti koji niz stanja je najverovatnije doveo do ovih rezultatima. Drugim rečima, interesuje nas $\arg\max_{x_{1:t}} \{\mathbb{P}(x_{1:t} \mid e_{1:t})\}$.
- Određivanje najverovatnijeg objašnjenja značajno u polju prepoznavanja govora ($E_{1:t}$ su zvuci, a $X_{1:t}$ su reči/glasovi) i rekonstrukciji niza bitova primljeno korišćenjem kanala sa puno šuma.
- Postupci će biti *iterativni*: postojaće rekurentna veza između procena koja će dati *tačnije* procene. Reč je o specijalnom primeru procesa *maksimizacije očekivanja* (*EM algoritam*).

Osnovna ideja filtriranja

- Filtriranje ćemo raditi *online*: kako dolaze rezultati obzervacija, dolaze i nove procene. Dakle, tražimo neku funkciju f za procenu:

$$\mathbb{P}\left(X_{t+1} \mid e_{1:t+1}\right) = f\left(e_{t+1}, \mathbb{P}\left(X_t \mid e_{1:t}\right)\right)$$

- Dva dela računice: najpre se aktuelna raspodela pomeri "unapred" sa t na $t+1$ (predviđanje!); nakon toga, ažurira se korišćenjem nove obzervacije E_{t+1}.
- Kraći račun:

$$\begin{aligned}
\mathbb{P}\left(X_{t+1} \mid e_{1:t+1}\right) &= \mathbb{P}\left(X_{t+1} \mid e_{1:t}, e_{t+1}\right) \\
&= \frac{\mathbb{P}\left(e_{t+1} \mid e_{1:t}, X_{t+1}\right) \mathbb{P}\left(X_{t+1} \mid e_{1:t}\right)}{\mathbb{P}\left(e_{1:t+1}\right)} \\
&= \frac{\mathbb{P}\left(e_{t+1} \mid X_{t+1}\right) \mathbb{P}\left(X_{t+1} \mid e_{1:t}\right)}{\mathbb{P}\left(e_{1:t+1}\right)}
\end{aligned}$$

Forward iteracija

- Dakle, ako je α *konstanta normalizacije*:

$$\mathbb{P}\left(X_{t+1} \mid e_{1:t+1}\right) = \alpha \mathbb{P}\left(e_{t+1} \mid X_{t+1}\right) \mathbb{P}\left(X_{t+1} \mid e_{1:t}\right)$$

- Jedino je pitanje izračunavanja $\mathbb{P}\left(X_{t+1} \mid e_{1:t}\right)$:

$$\mathbb{P}\left(X_{t+1} \mid e_{1:t}\right) = \sum_{x_t} \mathbb{P}\left(X_{t+1} \mid x_t, e_{1:t}\right) \mathbb{P}\left(x_t, e_{1:t}\right)$$

- Na osnovu Markovljevog svojstva, konačno dobijemo:

$$\mathbb{P}\left(X_{t+1} \mid e_{1:t+1}\right) = \alpha \mathbb{P}\left(e_{t+1} \mid X_{t+1}\right) \sum_{x_t} \mathbb{P}\left(X_{t+1} \mid x_t\right) \mathbb{P}\left(x_t, e_{1:t}\right)$$

- Ovo je osnova za iteraciju: ako je $f_{1:k}$ **(oprez sa notacijom!)** aktuelna procena za $\mathbb{P}\left(X_t \mid e_{1:k}\right)$, možemo da dobijemo "novu" procenu:

$$f_{1:k+1} = \alpha \textsc{Forward}\left(f_{1:k}, e_{k+1}\right)$$

Primer filtriranja

- Neka inicijalno smatramo da je podjednaka verovatnoća da se Artur odmara i da se se zabavlja: 0.5.
- U prvom satu nije bilo eksplozije. Najpre vršimo predviđanje:
$$\langle 0.9, 0.1\rangle \cdot 0.5 + \langle 0.3, 0.7\rangle \cdot 0.5 = \langle 0.5, 0.5\rangle$$
a onda, uz činjenicu da nije bilo eksplozije:
$$\alpha \langle 0.6, 0.95\rangle \langle 0.5, 0.5\rangle = \alpha \langle 0.3, 0.475\rangle \approx \langle 0.39, 0.61\rangle$$
- U drugom satu opet nije bilo ekspozije. Najpre vršimo predviđanje:
$$\langle 0.9, 0.1\rangle \cdot 0.39 + \langle 0.3, 0.7\rangle \cdot 0.61 \approx \langle 0.53, 0.47\rangle$$
a onda, uz činjenicu da nije bilo eksplozije:
$$\alpha \langle 0.6, 0.95\rangle \langle 0.53, 0.47\rangle \approx \alpha \langle 0.32, 0.45\rangle \approx \langle 0.42, 0.58\rangle$$

Predviđanje

- Predviđanje je samo filtriranje bez dodavanja novih obzervacija.
- Može da se pokaže da će predviđanje, uz veliku vrednost k, *konvergirati stacionarnoj raspodeli* Markovljevog lanca. Da li je ovo očekivan rezultat?

Problem učenja

- Podsetnik: za uočene izlaze $e_1, e_2, \ldots, e_t$, odrediti parametre skrivenog Markovljevog modela tako da se maksimizuje $\mathbb{P}\left(e_{1:t} \mid \text{parametri HMM}\right)$.
- Uočen niz izlaza koji se koristi za određivanje parametara HMM se zove **trening niz**. Problemi učenja su ključni jer nam omogućavaju da napravimo "najbolje moguće" modele u stvarnim procesima.

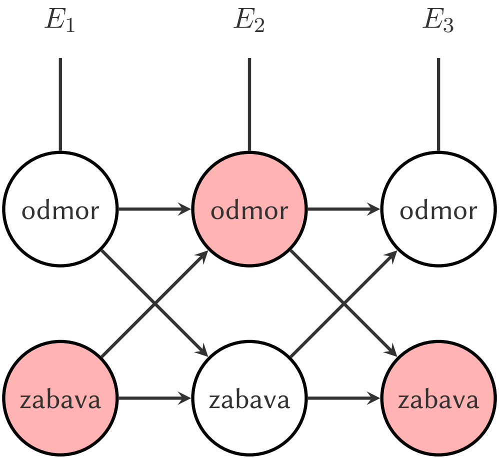
Rešenje za učenje

- Za ovo se koriste razne iterativne procedure.
- Metod koji se koristi u praksi je **Baum-Welch**.

Interpretacija najverovatnijeg objašnjenja

- Određivanje najverovatnijeg objašnjenja je možda najzanimljiviji od svih ovih problema.
- Zašto ne možemo samo da primenimo *forward-backward* algoritam za uviđanje i onda konstruišemo niz stanja?
- Za rešavanje ovog problema koristimo **Viterbijev algoritam**, jedan karakterističan primer *dinamičkog programiranja*. Složenost ovog algoritma će biti *linearna*.
- Svaki niz stanja ćemo interpretirati kao *put u grafu* čiji su čvorovi sva moguća stanja u nekom trenutku.

Viterbijev algoritam: primer puta



Viterbijev algoritam: iteracija

- Najpre je neophodno primetiti:

$$\max_{x_{1:t}} \{\mathbb{P}(x_{1:t}, X_{t+1} \mid e_{1:t+1})\} =$$

$$\alpha \mathbb{P}(e_{t+1} \mid X_{t+1}) \max_{x_t} \left\{ \mathbb{P}(X_{t+1} \mid x_t) \max_{x_{1:t-1}} \{\mathbb{P}(x_{1:t-1}, x_t \mid e_{1:t})\} \right\}$$

- Ova jednačina je veoma slična jednačini filtriranja! Ovde je samo neophodno uvesti oznaku za verovatnoće najverovatnijeg puta to svakog stanja x_t:

$$m_{1:t} = \max_{x_{1:t-1}} \{\mathbb{P}(x_{1:t-1}, X_t \mid e_{1:t})\}$$

i onda imamo:

$$m_{1:t+1} = \alpha \mathbb{P}(e_{t+1} \mid X_{t+1}) \max_{x_t} \{\mathbb{P}(X_{t+1} \mid x_t) m_{1:t}\}$$

Viterbijev algoritam: zaključak

- Nakon što se odrede sve vrednosti m, nije problem rekonstruisati niz.
- Viterbijev algoritam je izuzetno primenljiv. Čest primer je prepoznavanje govora, ali je najprimenljiviji rezultat u teoriji skrivenih Markovljevih lanaca.
- Postoji i matrični zapis, ali zahteva neke izmene koje su van opsega ovog predavanja.

Pregled skrivenih Markovljevih modela

- Razmatrali smo stacionarne procese koji poštuju Markovljevo svojstvo, odnosno procese nezavisne od prošlosti.
- Videli smo šta je model prelaza, a šta senzorski model.
- Razmotrili smo probleme filtriranja, predviđanja, uviđanja, najverovatnijeg objašnjenja. Bavili smo se detaljno FORWARD iteracijom i analizirali Viterbijev algoritam.
- Stekli smo uvid u pojam skrivenih Markovljevih modela i kako se ovi konkretni problemi interpretiraju u ovom kontekstu.

Probabilističke metode i mašinsko učenje

Zaključivanje nad multipleks mrežama

Petar Veličković

Matematička gimnazija
Nedelja informatike v2.0

16. decembar 2015.

Zašto multipleks mreže?

Na samom kraju, odradićemo kratak uvod u jednu od aktuelnijih vrsta kompleksnih mreža, i njene primene u mašinskom učenju.

- **Multipleks mreže** (*multiplex networks*) su bile centralna tema mog diplomskog rada na Kembridžu—ovo je kao rezultat imalo objavu naučnog rada (*Journal of Complex Networks, Oxford*).
- Ove mreže poseduju izuzetan potencijal za modeliranje mnogih sistema iz stvarnog života.
- Moj rad je prvi pokušaj razvoja algoritma mašinskog učenja nad ovim mrežama, i rezultati su bili veoma zadovoljavajući!

Roadmap

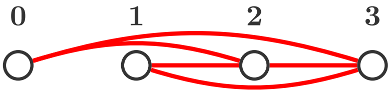
1. Počećemo sa nekoliko slajdova koji (ne)formalno definišu multipleks mreže i par motivišućih primera.

2. Nakon toga ćemo se ukratko osvrnuti na to kako, koristeći rešenja za standardne HMM probleme, možemo praktično koristiti HMM za neki problem mašinskog učenja.

3. Na samom kraju, pokazaćemo kako se ova dva koncepta mogu integrisati (tj. kako sam ih ja integrisao...).

Počnimo sa grafovima!

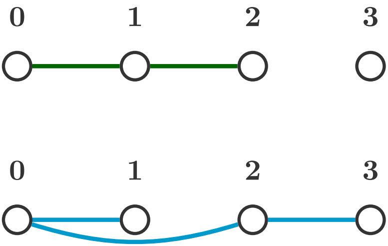
- Zamislite da imate četiri čvora, i zaključili ste da su određeni parovi njih na neki način *povezani*.



- Dobili ste vaš uobičajen, dosadan **graf**; u ovom kontekstu uglavnom nazivan *monopleks mrežom*.

Još neki grafovi

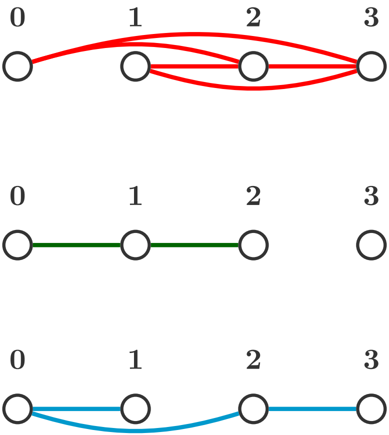
- Sada ste primetili da, u nekim drugim “referentnim sistemima” (videćemo primere uskoro), ove čvorove možemo povezivati na druge načine.



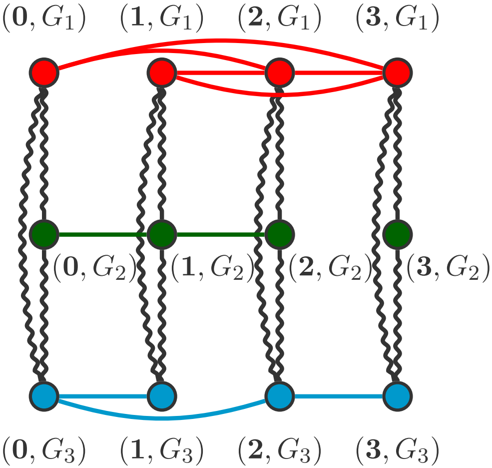
Uticaj

- Konačno, zaključili ste da ovi "slojevi interakcija" nisu nezavisni jedan od drugog, nego mogu međusobno interagovati na netrivijalne načine (formirajući tako *"mrežu mreža"*).
- Multipleks mreže nam daju relativno jednostavan način da predstavimo ove interakcije—dodavanjem novih *međuslojnih ivica* između čvorova i njihovih "slika" u drugim slojevima.
- Ako se vratimo na prethodni primer…

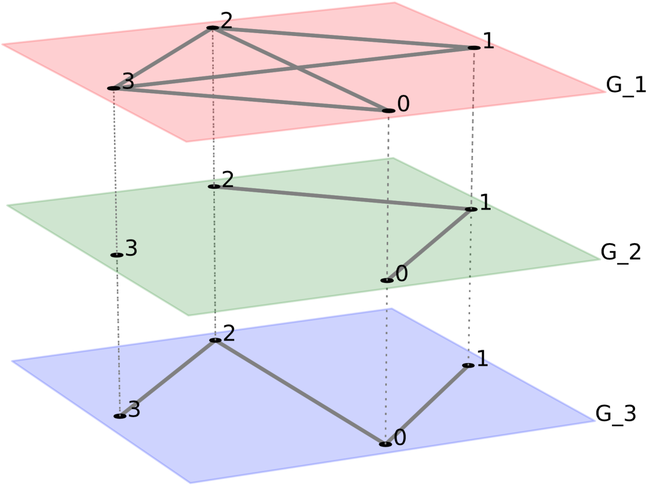
Prethodni primer



Prethodni primer



Napravili smo multipleks mrežu!



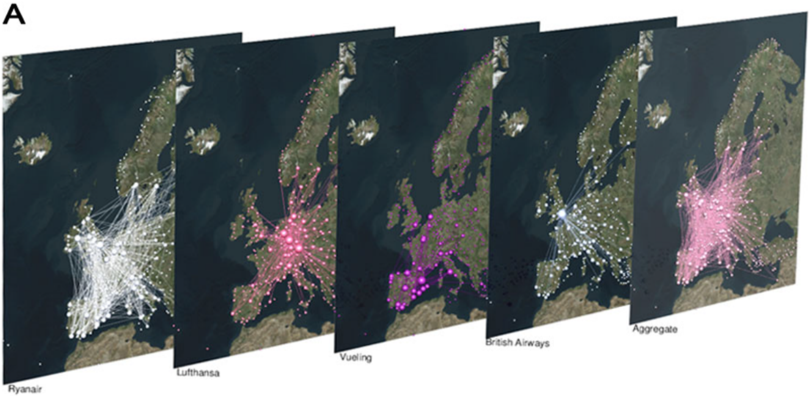
Primeri

Iako je ovo jednostavan model, za brojne stvarne sisteme se ispostavilo da iskazuju "multipleks" ponašanje. Primeri uključuju:

Primeri

Iako je ovo jednostavan model, za brojne stvarne sisteme se ispostavilo da iskazuju “multipleks” ponašanje. Primeri uključuju:

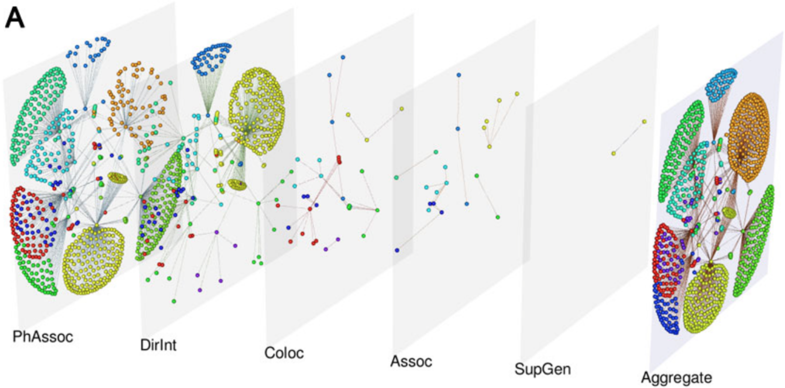
- Transportne mreže (De Domenico *et al.*)



Primeri

Iako je ovo jednostavan model, za brojne stvarne sisteme se ispostavilo da iskazuju “multipleks” ponašanje. Primeri uključuju:

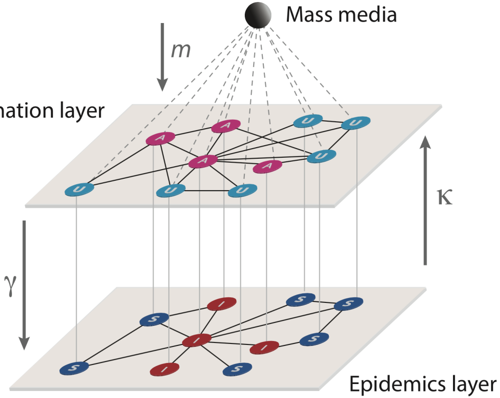
- Genetičke mreže (De Domenico *et al.*)



Primeri

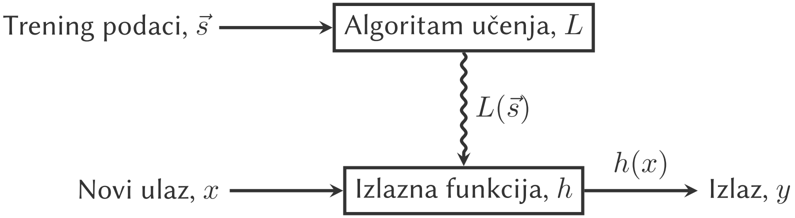
Iako je ovo jednostavan model, za brojne stvarne sisteme se ispostavilo da iskazuju "multipleks" ponašanje. Primeri uključuju:

- Socijalne mreže (Granell *et al.*)



Supervizirano učenje

- Jedan od najčešćih vidova mašinskog učenja je *supervizirano učenje*; kada je potrebno, posmatrajući neki set podataka sa poznatim izlazima (*trening podatke*) konstruisati funkciju koja zaključuje izlaze za do tada neviđene ulaze (*test podatke*).



Binarna klasifikacija

- Najjednostavniji primer problema koji se može rešiti superviziranim učenjem je *binarna klasifikacija*: za neke dve klase (C_1 i C_2), odrediti kojoj klasi pripada neki ulaz x.
- Primene su **višestruke**:
  - *Dijagnostika* (da li pacijent boluje od neke bolesti, na osnovu nivoa šećera u krvi, krvnog pritiska, i sl. merenja?);
  - *Kreditiranje* (da li se od osobe može očekivati povraćaj kredita, na osnovu njene finansijske istorije?);
  - *Trgovina* (da li kupiti ili prodati neku akciju, na osnovu prethodnih pomeranja njene cene?);
  - *Automatska vožnja* (da li su uslovi za vožnju opasni, na osnovu meteoroloških merenja?);
  - ...

Klasifikacija sa HMM-ovima

- Trening podaci nam se sastoje od sekvenci za koje *znamo* da li su u klasi C_1 ili C_2.
- Možemo izgraditi *dva odvojena HMM-a*; jedan proizvodi sekvence iz C_1, a drugi proizvodi sekvence iz C_2. Modele ćemo trenirati tako što pustimo dovoljan broj iteracija *Baum-Welch algoritma* nad onim sekvencama iz trening podataka koje pripadaju njihovim klasama.
- Nakon izgradnje modela, klasifikacija novih sekvenci je jednostavna; korišćenjem *forward algoritma* odredimo da li je verovatnije da je nova sekvenca proizvedena od strane C_1 modela ili od strane C_2 modela.

Malo drugačiji problem

- Pretpostavimo da imamo pristup *više* od jednog izlaza u bilo kom momentu (npr. ako merimo parametre nekog pacijenta kroz vreme, možemo u isto vreme izmeriti i krvni pritisak i nivo šećera u krvi).

- Možemo probati da preformulišemo našu izlaznu matricu $\mathbf{O}$, tako da ima $k+1$ dimenzija za k izlaza:

$$\mathbf{O}_{x,a,b,c,\dots} = \mathbb{P}\left(a, b, c, \dots | x\right)$$

…ali ovo ubrzo postaje nepraktično, i sa memorijske i sa numeričke strane (pogotovo ukoliko su izlazi iz $\mathbb{R}$)…

Modeliranje

- Postoji više načina za rukovanje višestrukim izlazima, međutim većina ne uzima u obzir potencijalnu netrivijalnost veza između ovih izlaza.
- Tu na scenu stupaju *multipleks mreže*, kao model koji se pokazao kao efikasan u modeliranju realnih sistema.
- **Osnovna ideja:** modeliraćemo *svaki izlaz zasebno* unutar jednog HMM-a, pa ćemo onda uvezati te HMM-ove u jedan veći *multipleks HMM*. Ova celokupna struktura će se i dalje ponašati kao HMM, pa ćemo moći da klasifikujemo koristeći *forward algoritam*, kao i ranije.

Međuslojne ivice

- Dakle, pretpostavljamo da imamo k HMM-ova (po jedan za svaki izlaz) sa n čvorova.
- U svakom trenutku, sistem se nalazi u jednom čvoru jednog HMM-a, i može:
  - promeniti čvor unutar istog HMM-a, ili
  - promeniti HMM (a ostati u istom čvoru).
- Verovatnoće promene HMM-a u svakom koraku možemo karakterisati jednom matricom, $\boldsymbol{\omega}$ (veličine $k \times k$).
- Tako $\boldsymbol{\omega}_{ij}$ daje verovatnoću da u bilo kom koraku pređemo iz HMM-a koji proizvodi i-ti izlaz u HMM koji proizvodi j-ti izlaz.

Parametri multipleks HMM-a

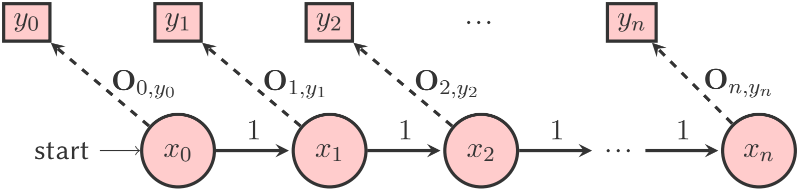
- **Važno:** kada smo u i-tom HMM-u zanima nas samo verovatnoća proizvodnje i-tog izlaza; *ne razmatramo preostalih* $k-1$ *izlaza! (smatramo da su proizvedeni sa verovatnoćom 1)*
- Sa tim na umu, ceo sistem možemo posmatrati kao HMM nad stanjima (x, i), gde je x čvor a i tip izlaza, i parametrima:

$$\pi_{(x,i)} = \boldsymbol{\omega}_{ii}\pi_x^i$$

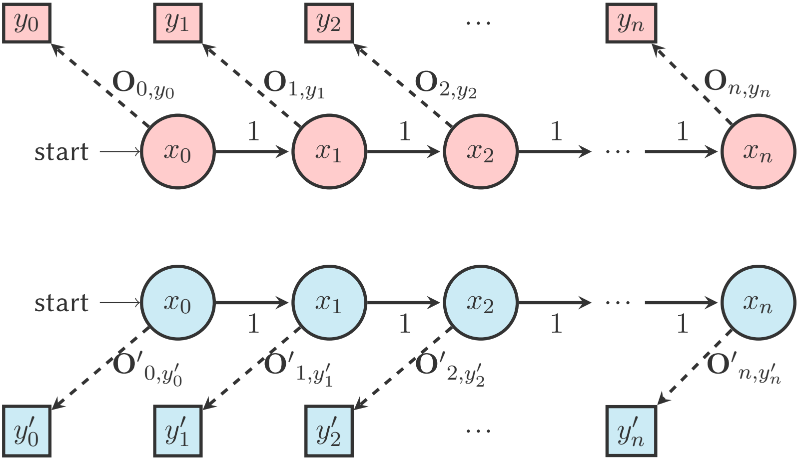
$$\mathbf{T}_{(a,i),(b,j)} = \begin{cases} 0 & a \neq b \wedge i \neq j \\ \boldsymbol{\omega}_{ij} & a = b \wedge i \neq j \\ \boldsymbol{\omega}_{ii}\mathbf{T}_{ab}^i & i = j \end{cases}$$

$$\mathbf{O}_{(x,i),\vec{y}} = \mathbf{O}_{xy_i}^i$$

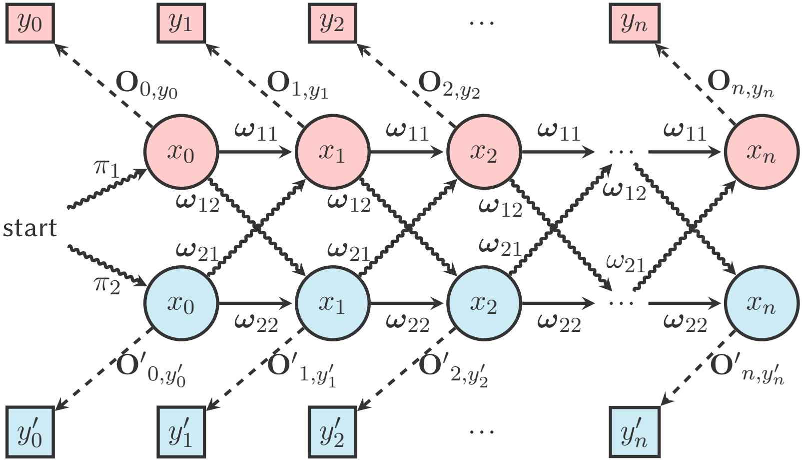
Primer: HMM lanac



Primer: Dva HMM lanca



Primer: Multipleks HMM lanac



Trening

- Treniranje svakog HMM-a u multipleksu zasebno (tj. ordeđivanje parametara $\vec{\pi}^i, \mathbf{T}^i, \mathbf{O}^i$) se može raditi kao i ranije (koristeći *Baum-Welch* algoritam).
- Određivanje matrice $\boldsymbol{\omega}$ je daleko *teži* zadatak; ova matrica definiše relativne zavisnosti između izlaznih podataka, što za mnoge praktične probleme uopšte ne možemo odrediti!
- Stoga, pribegavamo metodama optimizacije koje ne prave dodatne pretpostavke o problemu koji rešavamo—u mom radu, korišćen je *genetički algoritam* za određivanje optimalne vrednosti $\boldsymbol{\omega}$.

Klasifikacija

- Kao što je već rečeno, za binarnu klasifikaciju:
  - Pravimo dva odvojena modela;
  - Svaki model treniramo nad onim sekvencama iz trening podataka koje mu pripadaju;
  - Klasifikujemo nove podatke u onu klasu za čiji model *forward algoritam* da veću verovatnoću.
- U mom radu, ovaj metod je primenjen na klasifikaciju pacijenata za *rak dojke* na osnovu dva različita genetička merenja nad genima za koje se sumnja da su odgovorni za pojavu raka: postignuta tačnost $\mathbf{94.2\%}$ na test podacima!

Implementacija i disertacija

- Kompletnu implementaciju ovog modela (u jeziku C++) možete naći na: `https://github.com/PetarV-/Ariana`.
- Moj diplomski rad (gde možete, između ostalog, naučiti detalje vezane za ocenjivanje efikasnosti modela) možete naći na: `https://www.dropbox.com/s/2hm79le1uk0ps00/pv273%20-%20Molecular%20multiplex%20network%20inference.pdf?dl=0`

Kraj!

- Stigli smo, konačno, do kraja današnjeg “dana verovatnoće”!
- Nadamo se da ste uspeli da zadržite pažnju, a verujemo da su današnje ideje bile nove i izazovne.
- Ovo je bio samo jedan (od mnogih) uvoda u vrlo široku i bogatu oblast mašinskog učenja! Nadamo se da ćemo moći da se ovom temom bavimo i narednih godina.
- Ponavljamo da je današnji dan bio *eksperiment*.
  Očekujemo detaljne reakcije!