

SORMENJÄLJET (fingerprints)

- ★ Pitkät avaimet voidaan kuvata lyhyemmille *sormenjäljille*
- ★ Nyt siis ei säilytetä avaimia lokeroissa, vaan tiedossa on vain sormenjäljet
- ★ Yhteentörmäysten takia hajautuksessa tapahtuu virheitä
- ★ Olk. $S = \{s_1, \dots, s_m\}$ suuren universumin U joukko
- ★ Haluamme esittää alkioita tilaa säästämällä s.e. voimme tehokkaasti vastata kysymyksiin $x \in S$?
- ★ Sallimme virheellisen vastauksen pienellä tn.:llä
- ★ Montako bittiä b tulisi sormenjälkiin käyttää, jotta virhetn. pysyy matalana

145

- ★ Tn., että alkioilla $x \notin S$ on eri sormenjälki kuin tietyllä $s \in S$ on $(1 - 1/2^b)$
- ★ Virhetn. kaikkiaan on siis:

$$1 - (1 - 1/2^b)^m \geq 1 - e^{-m/2^b}$$
- ★ Jos haluamme virhetn.:n olevan vaaka c pienemmän, tulee vaatia

$$e^{-m/2^b} \geq 1 - c, \text{ josta seuraa}$$

$$b \geq \log_2 \frac{m}{\ln(1/(1 - c))}$$
- ★ Eli nyt tarvitaan $b = \Omega(\log_2 m)$ bittiä
- ★ Jos toisaalta käytämme $2 \log_2 m$ bittiä, niin virhetn. putoaa arvoon

$$1 - \left(1 - \frac{1}{m^2}\right)^m < \frac{1}{m}$$
- ★ Jos esim. $m = 2^{16} = 65\,536$, niin 32:n bitin sormenjäljillä saavutetaan virhetn., joka on pienempi kuin $1/65\,536$

146

BLOOM-FILTTERIT

- ★ Bloom-filtteri on n :n bitin taulukko $A[0, \dots, n - 1]$
- ★ Lähtötilanteessa $A[i] = 0$ kaikilla i
- ★ Käytetään k :ta riippumatonta hajautusfunktioita $h_1, \dots, h_k: U \rightarrow \{0, \dots, n - 1\}$
- ★ Oletetaan funktioiden hajauttavan alkioita tasaisesti joukolle $\{0, \dots, n - 1\}$
- ★ Esitettävän joukon $S \subseteq U$ alkiolla $s \in S$ bitit $A[h_i(s)]$ asetetaan arvoon 1 kaikilla $1 \leq i \leq k$
- ★ Jos $A[h_i(x)] = 0$ jollakin i , niin $x \notin S$
- ★ Toisaalta, jos $A[h_i(x)] = 1$ kaikilla i , niin on mahdollista että $x \notin S$, vaikkakin joudumme olettamaan että $x \in S$

147

- ★ Kun kaikki S :n m alkiota on hajautettu Bloom-filtteriin on tn., että tietty bitti on vielä arvoltaan 0

$$\left(1 - \frac{1}{n}\right)^{km} \approx e^{-km/n}$$

- ★ Merk. $p = e^{-km/n}$
- ★ Oletetaan nyt, että kun kaikki alkioita on hajautettu, on 0-bittien osuus p
- ★ Virhetn. on siis

$$\begin{aligned} \left(1 - \left(1 - \frac{1}{n}\right)^{km}\right)^k &\approx (1 - e^{-km/n})^k \\ &= (1 - p)^k \end{aligned}$$

- ★ Merk. $f = (1 - p)^k$

148

- ★ Olk. m ja n annettu ja tarkoituksemme optimoida hajautusftioiden lkm k virheen tn.:n f minimoimiseksi
 - ◊ Useampien ftioiden käyttäminen antaa suuremman mahdollisuuden löytää 0-bitti alkiolle $\notin S$
 - ◊ Toisaalta harvempien ftioiden käyttö lisää 0-bittien osuutta taulukossa
- ★ Olk. $f = e^g$, eli $g = k \ln(1 - e^{-km/n})$
- ★ Virhetn.:n f minimoiminen on ekvivalenttia g :n minimoimisen kanssa k :n suhteen

$$\frac{dg}{dk} = \ln(1 - e^{-km/n}) + \frac{km}{n} \frac{e^{-km/n}}{1 - e^{-km/n}}$$
- ★ Derivaatta saa arvon 0 kun $k = (\ln 2)(n/m)$ ja tämä piste on globaali minimi

149

- ★ Tällöin $f = (1/2)^k \approx (0,6185)^{n/m}$
- ★ Virhetn. siis pienenee eksponentiaalisesti osamäärän n/m — alkiota kohti käytettyjen bittien lkm:n — suhteen
- ★ Bloom-filtteri on sormenjälkitekniikan yleistys
- ★ Vakiovirhetn.:n saavuttamiseksi kunkin alkion sormenjälkeä varten tarvitaan $\Omega(\log m)$ bittiä, joka voi olla käytännön kannalta liikaa
- ★ Bloom-filtterillä päästään vakiovirhetn.:n pitämällä suhde n/m vakiona, esim. $n = cm$ pienellä vakiolla c
- ★ Kun esim. $c = 8$ ja $k = 5$ tai $k = 6$, niin virhetn. on n. 0,02
- ★ Sormenjälkiä oleellisesti pienemmällä tilankäytöllä saavutetaan hyvä virhetn.

150

- ★ Virhetn. f on p :n funktiona

$$f = (1 - p)^k = (1 - p)^{(-\ln p)(m/n)}$$

$$= (e^{-\ln(p) \ln(1-p)})^{m/n}$$
- ★ Eksponentin symmetrisyyden perusteella on helppo nähdä, että arvo $p = 1/2$ minimoi virhetn.:n
- ★ Paras tulos siis saavutetaan hajautus-taulukon vaikuttaessa satunnaiselta bittijonolta (koska 0-bitin tn. on 1/2)
- ★ Luovutaan oletuksesta, että hajautuksen jälkeen 0-bittien osuus on p
- ★ Heitetään mk palloa n uurnaun, tyhjien urnien odotusarv. osuus on $p' = (1 - 1/n)^{km}$
- ★ Korollarin 5.9 (tai 5.11) perusteella poikkeaminen tästä odotusarvosta suurilla arvoilla n on epätn.

151

Satunnaisverkot

- ★ JTKT:sta tiedämme esim. verkko-ongelmien Hamiltonin kehä, riippumaton joukko ja solmupeite olevan laskennallisesti vaativia
- ★ Ovatko ongelmat vaativia useimmille vaiko vain harvoille verkoille?
- ★ Satunnaisverkot ovat probabilistinen malli tämänkaltaisten kysymysten tutkimiseksi
- ★ Mallissa $G_{n,p}$ tarkastellaan kaikkia suuntaamattomia n :n solmun v_1, \dots, v_n verkkoja
- ★ Verkon, jossa on tietyt m kaarta, tn. on

$$p^m (1 - p)^{\binom{n}{2} - m}$$

152

- ★ $G_{n,p}$ -verkko voidaan generoida tarkastelemalla kaikkia $\binom{n}{2}$ mahdollista kaarta järjestyksessä ja riippumattomasti lisäten kukin niistä verkkoon tn.:llä p
- ★ Kaarten lkm:n odotusarvo on $\binom{n}{2}p$ ja kunkin solmun odotusarvoinen aste on $(n-1)p$
- ★ Toisessa mallissa $G_{n,N}$ tutkitaan n :n solmun suuntaamattomia verkkoja, joissa on täsmälleen N kaarta
- ★ Mahd. verkkoja on kaikkiaan $\binom{n}{N}$
- ★ Tässä mallissa verkon generoimiseksi voidaan aloittaa kaarettomasta verkosta ja arpoa verkkoon N kaarta tasaisen jakauman mukaan ilman takaisinpanoa

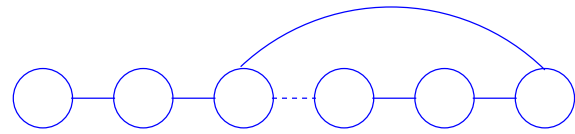
153

- ★ Mallien $G_{n,p}$ ja $G_{n,N}$ keskinäinen suhde on hieman kuin eksaktin ja Poisson-tapauksen suhde pallot ja uurnat -mallissa
- ★ Kun $p = N/\binom{n}{2}$, niin $G_{n,p}$ -satunnaisverkon kaarten lkm on keskittynyt arvon N ympäristöön
- ★ Ehdolla, että kaaria on tasan N , on tämä verkko yhtä suurella tn.:llä mikä tahansa mallin $G_{n,N}$ verkoista
- ★ Mallissa $G_{n,N}$ "heitetään" kaaria verkkoon
- ★ Kullakin kaarella on kaksi päätä, joten tavallaan heitetään kerrallaan kaksi toisiinsa liittyvää palloa
- ★ Silti pallot ja uurnat -malli soveltuu satunnaisverkkojen analyysiin

154

HAMILTONIN KEHÄT

- ★ Hamiltonin kehä (HC) on kussakin verkon solmussa kerran käyvä kehä
- ★ HC:n löytäminen on NP-täydellinen ongelma
- ★ Satunnaisalgoritmillä HC voidaan kuitenkin löytää tehokkaasti joillekin satunnaisverkoille
- ★ Olk. G suuntaamaton verkko, $P = v_1, \dots, v_k$ yksinkertainen polku G :ssä ja (v_k, v_1) kaari G :ssä
- ★ Tällöin yksinkertainen polku G :ssä on myös $P' = v_1, \dots, v_i, v_k, v_{k-1}, \dots, v_{i+1}$
- ★ P' on P :n rotaatio ja kaari (v_k, v_1) rotaatiokaari



155

HC-ALGORITMI

Syöte: $G = (V, E)$, jossa on n solmua.

Tulos: HC tai epäonnistuminen.

1. Vedä satunnainen solmu polun pääksi.
2. Toista kunnes (rotaatiokaari sulkee HC:n) tai (polun pään käyttämättömien kaarten lista on tyhjä):
 - (a) Olk. nykyinen polku $P = v_1, \dots, v_k$, missä v_k on polun pää, ja olk. (v_k, u) ensimmäinen kaari pään listassa.
 - (b) Poista (v_k, u) v_k :n ja u :n listasta.
 - (c) Jos $u \neq v_i$ kaikilla $1 \leq i \leq k$, niin lisää $u = v_{k+1}$ polun pääksi.
 - (d) Muutoin ($u = v_i$) rotatoi nykyinen polku kaaren (v_k, v_i) suht., polun pää on nyt v_{i+1} . (Jos $k = n$ ja valitaan (v_n, v_1) , niin HC sulkeutuu).
3. Palauta löytynyt HC tai ilmoita epäonnistumisesta.

156

- ★ Ed. algoritmi on liian vaativa analysoitavaksi
- ★ Ongelman muodostavat ehdolliset riippuvuudet, jotka seuraavat kun kaaria katsotaan solmujen vieruslistoista
- ★ Muokattu algoritmi ylläpitää kahta listaa kutakin solmua kohti
 - ◇ $KÄY(v)$ on algoritmin silloin kun v on ollut polun päässä käyttämät v :hen liittyvät kaaret
 - ◇ $EI-KÄY(v)$ on muut v :hen liittyvät kaaret
- ★ Analysoimme ensin algoritmin olettaen, että jokainen mahdollisesta v :hen liittyvästä $n - 1$ kaaresta on aluksi listassa $EI-KÄY(v)$ riippumattomasti jollain tn.:llä q

157

- ★ Lisäksi kaaret ovat listassa satunnaisessa järjestyksessä
- ★ Ennen algoritmin käynnistämistä verkko G generoidaan lisäämällä kunkin solmun v listaan $EI-KÄY(v)$ jokaisen mahdollisen kaaren (v, u) tn.:llä q
- ★ Täten kaari (v, u) voi olla solmun v listassa, mutta ei u :n listassa
- ★ Kun kaari (v, u) poistetaan polun päästä v listasta, se edelleenkin jää solmun u listaan (jos se siellä on)
- ★ Kussakin askelella valitsemalla rotaatiokaari sopivalla tn.:llä listoista $KÄY$ ja $EI-KÄY$ sekä kääntämällä polku sopivalla pienellä tn.:llä, saadaan polun päästä valinta tapahtumaan tasaisen jakauman mukaan verkon kaikkien solmujen joukosta

158

MUOKATTU HC-ALGORITMI

Syöte: Verkko $G = (V, E)$, jossa on n solmua ja kullakin niistä kaksi kaarilistaa.

Tulos: HC tai epäonnistuminen.

1. Vedä satunnainen solmu polun pääksi.
2. Toista kunnes (rotaatiokaari sulkee HC:n) tai (polun päästä EI-KÄY-lista on tyhjä):
 - (a) Olk. nykyinen polku $P = v_1, \dots, v_k$, missä v_k on polun pää.
 - (b) Suorita yksi seuraavista kohdista:
 - i. Tn.:llä $1/n$: käännä polku s.e. v_1 :stä tulee polun pää.
 - ii. Tn.:llä $|KÄY(v_k)|/n$: valitse tasaisen jakauman mukaan kaari listasta $KÄY(v_k)$; jos se on (v_k, v_i) , niin rotatoi nykyinen polku kaaren (v_k, v_i) suht., polun pää on nyt v_{i+1} . (Jos kaari on (v_k, v_{k-1}) , niin ei tehdä mitään).
 - iii. Tn.:llä $1 - 1/n - |KÄY(v_k)|/n$: valitse ensimmäinen kaari (v_k, u) listasta $EI-KÄY(v_k)$.
 - A. Jos $u \neq v_i$ kaikilla $1 \leq i \leq k$, niin lisää $u = v_{k+1}$ polun pääksi.
 - B. Muutoin ($u = v_i$) rotatoi nykyinen polku kaaren (v_k, v_i) suht., polun pää on nyt v_{i+1} . (Jos $k = n$ ja valitaan (v_n, v_1) , niin HC sulkeutuu).
 - (c) Päivitä $KÄY$ - ja $EI-KÄY$ -listat.
3. Palauta löytynyt HC tai ilmoita epäonnistumisesta.

159

160

- ★ Polun pään voi kussakin askelessa ajatella olevan tasaisen jakauman mukaan vedetty kaikkien verkon solmujen joukosta riippumatta historiasta:

Lemma 5.15: *Olk. verkko generoitu edellä kuvatulla tavalla ja muokattua HC-algoritmia sovelletaan siihen. Olk. V_t polun pää t :n askelen jälkeen. Kunhan t :nessä askelessa on ainakin yksi ei-käytetty kaari polun pään listassa, niin mille tahansa solmulle u pätee*

$$\Pr(V_{t+1} = u \mid V_t = u_t, \dots, V_0 = u_0) = 1/n.$$

Todistus. Olk. polku $P = v_1, \dots, v_k$.

Solmusta v_1 voi tulla polun pää vain polku kääntämällä, joten algoritmin mukaisesti $V_{t+1} = v_1$ tn.:llä $1/n$.

161

Jos $u = v_{i+1} \in P$ ja $(v_k, v_i) \in \text{KÄY}(v_k)$, niin tn., että $V_{t+1} = u$ on

$$(|\text{KÄY}(v_k)|/n)(1/|\text{KÄY}(v_k)|) = 1/n.$$

Edellisten tapausten ulkopuolelle jää, että kaari valitaan joukosta $\text{EI-KÄY}(v_k)$. Tällöin vierussolmu on jakautunut tasaisesti yli jäljellä olevien $n - |\text{KÄY}(v_k)| - 1$ solmun viivästetyn valinnan periaatteen perusteella.

Alunperin listaan $\text{EI-KÄY}(v_k)$ valittiin kukin kaari tn.:llä q s.e. ne ovat satunnaisessa järjestyksessä. Tämä on sama kuin X :n vierussolmun valitseminen v_k :lle tasaisen jakauman mukaan ilman takaisinpanoa, missä s -muuttuja $X \sim B(n-1, q)$.

Koska v_k :n lista valittiin muista solmuista riippumatta, algoritmin historia ei kerro

162

mitään listassa vielä olevista kaarista ja viivästetyn valinnan periaate pätee. Täten mikä tahansa kaarista jota emme vielä ole nähneet listassa $\text{EI-KÄY}(v_k)$ voi yhtä suurella tn.:llä kiinnittyä mihin tahansa $n - |\text{KÄY}(v_k)| - 1$ jäljellä olevaan solmuun.

Jos $u = v_{i+1} \in P$ ja $(v_k, v_i) \notin \text{KÄY}(v_k)$, niin tn., että $V_{t+1} = u$ on tn., että kaari (v_k, v_i) valitaan $\text{EI-KÄY}(v_k)$:sta seuraavaksi rotaatiokaareksi, eli

$$\left(1 - \frac{1}{n} - \frac{|\text{KÄY}(v_k)|}{n}\right) \left(\frac{1}{n - |\text{KÄY}(v_k)| - 1}\right) = \frac{1}{n}.$$

Lopulta, jos $u \notin P$, niin tn., että $V_{t+1} = u$ on sama kuin tn., että kaari (v_k, u) valitaan listasta $\text{EI-KÄY}(v_k)$. Tn. on sama kuin edellä. \square

163

- ★ Huomaa, että seuraavassa ei oleteta satunnaisverkon sisältävän HC:ta
- ★ Lauseesta siis seuraa, että tällä tavoin generoitu satunnaisverkko sisältää suurella tn.:llä HC:n

Lause 5.16: *Olk. muokatun HC-algoritmin syöteverkko s.e. lähtötilanteessa $\text{EI-KÄY}(v)$ -lista sisältää kunkin kaaren (v, u) , $v \neq u$, riippumattomasti tn.:llä $q \geq 20 \ln n/n$. Tällöin algoritmi löytää HC:n toistolauseen $O(n \ln n)$ iteraatioissa tn.:llä $1 - O(1/n)$.*

Todistus. Jos algoritmi epäonnistui, toinen seuraavista tapahtui:

\mathcal{E}_1 : Algoritmi toimi $3n \ln n$ askelta ilman, että yksikään EI-KÄY -lista tyhjeni, muttei löytänyt HC:tä.

\mathcal{E}_2 : Ainakin yksi EI-KÄY -lista tyhjeni ensimmäisten $3n \ln n$ silmukan iteraation kuluessa.

164

Lemma 5.15 \Rightarrow tapahtuman \mathcal{E}_1 yhteydessä polun pää kullakin askelella on tasaisesti jakautunut verkon solmujen joukossa. Tapahtuman \mathcal{E}_1 rajoittamiseksi tarkastellaan siis tn.:ttä, että HC:n löytäminen vaatii yli $3n \ln n$ askelta kun polun pää valitaan tasaisen jakauman mukaan.

Tilanne on sama kuin kuponginkerääjän ongelma n :llä kupongilla. Tn., ettei tiettyä kuponkia ole löydetty $2n \ln n$ satunnaisten kupongin joukosta on

$$\left(1 - \frac{1}{n}\right)^{2n \ln n} \leq e^{-2 \ln n} = \frac{1}{n^2}.$$

Yhdisteen tn.:n perusteella tn., että on olem. kuponki, jota ei ole löydetty on $1/n$.

Näin ollen $2n \ln n$ askelen jälkeen jokainen solmu on ollut polun päässä (siis erityisesti on polulla) tn.:llä $1/n$.

165

Hamiltonin *polun* muuntaminen HC:ksi edellyttää polun sulkemista. Kullakin askelella tn., että polku sulkeutuu on $1/n$.

Tn., ettei polusta tule sykliä seuraavien $n \ln n$ iteraation kuluessa on

$$\left(1 - \frac{1}{n}\right)^{n \ln n} \leq e^{-\ln n} = \frac{1}{n}.$$

Kaikkiaan siis $\Pr(\mathcal{E}_1) \leq 2/n$.

$\Pr(\mathcal{E}_2)$:n rajoittamiseksi tarkastellaan tapahtumia:

\mathcal{E}_{2a} : Ainakin yhden solmun EI-KÄY-listasta poistettiin väh. $9 \ln n$ kaarta ensimmäisten $3n \ln n$ silmukan iteraation kuluessa.

\mathcal{E}_{2b} : Ainakin yhden solmun EI-KÄY-listassa oli alunperin vähemmän kuin $10 \ln n$ kaarta.

166

Jotta \mathcal{E}_2 tapahtuisi, täytyy joko \mathcal{E}_{2a} :n tai \mathcal{E}_{2b} :n tapahtua, joten

$$\Pr(\mathcal{E}_2) \leq \Pr(\mathcal{E}_{2a}) + \Pr(\mathcal{E}_{2b}).$$

Lemman 5.15 perusteella tn., että tietty solmu v on polun pää on $1/n$ riippumattomasti kullakin askelella. Olk. X se lkm, jonka v on polun päässä ensimmäisten $3n \ln n$ askelen kulussa. $X \sim B(3n \ln n, 1/n)$ on yläraja v :n EI-KÄY-listasta poistetuille alkiuille.

Lauseen 4.4(1) Chernoffin rajaa parametrein $\delta = 2$ ja $\mu = 3 \ln n$ käyttäen saamme

$$\Pr(X \geq 9 \ln n) \leq \left(\frac{e^2}{27}\right)^{3 \ln n} \leq \frac{1}{n^2}.$$

Yhdisteen tn.:n perusteella $\Pr(\mathcal{E}_{2a}) \leq 1/n$.

167

Olk. Y solmun EI-KÄY-listan kaarten lkm lähtötilanteessa. Sen odotusarvo on $(n-1)q \geq 20(n-1) \ln n/n \geq 19 \ln n$ riittävän suurilla arvoilla n .

Lauseen 4.5(2) Chernoffin rajan perusteella

$$\begin{aligned} \Pr(Y \leq 10 \ln n) &\leq e^{-19 \ln n (9/19)^2 / 2} \\ &\leq \frac{1}{n^2}. \end{aligned}$$

Yhdisteen tn.:n perusteella tn., että yhdelläkään solmulla on liian vähän siihen liittyviä kaaria on kork. $1/n$. Ts. $\Pr(\mathcal{E}_{2b}) \leq 1/n$, joten $\Pr(\mathcal{E}_2) \leq 2/n$.

Kaikkiaan siis tn., ettei algoritmi löydä HC:tä $3n \ln n$ iteraatioissa on kork.

$$\Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2) \leq \frac{4}{n}.$$

□

168

- ★ Lopuksi vielä sovelletaan HC-algoritmia mallin $G_{n,p}$ satunnaisverkkoihin

Korollaari 5.17: *Kun EI-KÄY-listat alustetaan sopivasti, niin muokattu HC-algoritmi löytää mallista $G_{n,p}$ satunnaisesti valitusta verkosta HC:n tn.:llä $1 - O(1/n)$ kunhan $p \geq 40 \ln n/n$.*