Kapitel 4. Chernoffs Schranke und Anwendungen

Satz (Chernoffs Schranke I)

Seien $X_1, \dots, X_n$ unabhängige Zufallsvariablen; $X_i \in \{0,1\}$

$\text{Prob}[X_i = 1] = p_i$ mit $0 < p_i < 1$

Sei $X = \sum_{i=1}^{n} X_i$, $\mu = E[X] = p_1 + \dots + p_n$

Dann gilt für jedes $\delta > 0$:

$$\text{Prob}[X > (1+\delta)\cdot\mu] < \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}$$

Beweis: Sei $t \in \mathbb{R}$, $t > 0$

$$\rightarrow \text{Prob}[X > (1+\delta)\cdot\mu] = \text{Prob}\left[e^{t\cdot X} > e^{t\cdot(1+\delta)\cdot\mu}\right]$$

Markov-Ungl. $\longrightarrow$

$$\text{Prob}[X > (1+\delta)\cdot\mu] < \frac{E[e^{t\cdot X}]}{e^{t\cdot(1+\delta)\cdot\mu}}$$

$$E[e^{t\cdot X}] = E\left[e^{t\cdot\sum_{i=1}^{n} X_i}\right] = E\left[\prod_{i=1}^{n} e^{t\cdot X_i}\right]$$

$X_1, \dots, X_n$ unabhängig $\rightarrow$

$e^{t\cdot X_1}, \dots, e^{t\cdot X_n}$ unabhängig.

$$\longrightarrow \text{Prob}[X > (1+\delta)\mu] <$$

$$\frac{E[\prod_{i=1}^{n} e^{t \cdot X_i}]}{e^{t\cdot(1+\delta)\cdot\mu}} = \frac{\prod_{i=1}^{n} E[e^{t\cdot X_i}]}{e^{t\cdot(1+\delta)\cdot\mu}}$$

$$= \frac{\prod_{i=1}^{n} (p_i e^t + 1 - p_i)}{e^{t\cdot(1+\delta)\cdot\mu}} = \frac{\prod_{i=1}^{n} (1 + p_i(e^t - 1))}{e^{t\cdot(1+\delta)\cdot\mu}}$$

Mit $1 + x \leq e^x$ für $x = p_i(e^t - 1) > 0$

erhalten wir:

$$\text{Prob}[X > (1+\delta)\cdot\mu] <$$

$$\frac{\prod_{i=1}^{n} e^{p_i(e^t-1)}}{e^{t\cdot(1+\delta)\cdot\mu}} = \frac{e^{(e^t-1)\cdot\mu}}{e^{t\cdot(1+\delta)\cdot\mu}} \quad (*)$$

$E[X] = p_1 + p_2 + \dots + p_n$

(*) gilt für jeden $t > 0$

Sei $t = \ln(1+\delta)$

$$\longrightarrow \text{Prob}[X > (1+\delta)\cdot\mu] <$$

$$\frac{e^{\delta\cdot\mu}}{(1+\delta)^{(1+\delta)\cdot\mu}} = \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu} \qquad \square$$

Beachte: $\delta > 0 \rightarrow e^{\delta} < (1+\delta)^{(1+\delta)}$

D.h. Chernoffs Schranke liefert eine obere Schranke < 1.

Korollar: Seien die Voraussetzungen für Chernoffs Schranke I gegeben.

Sei weiter $\delta \geq 2e - 1$.

$$\Rightarrow \text{Prob}[X > (1+\delta)\cdot\mu] < 2^{-(1+\delta)\cdot\mu}$$

Beweis:

Chernoffs Schranke $\overset{I}{\longrightarrow}$

$$\text{Prob}[X > (1+\delta)\cdot\mu] < \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}$$

Aus $\delta > 2e-1$ folgt

$$\left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu} \leq \frac{e^{(1+\delta)\cdot\mu}}{(1+\delta)^{\mu(1+\delta)}} <$$

$$\frac{e^{(1+\delta)\cdot\mu}}{(2e)^{(1+\delta)\cdot\mu}} = 2^{-(1+\delta)\cdot\mu}$$ □

Satz (Chernoffs Schranke II)

Seien die Voraussetzungen für Chernoffs Schranke I gegeben und sei weiter $0 < \delta \leq 1$ beliebig. →

$$\text{Prob}[X < (1-\delta)\cdot\mu] < e^{-\mu\cdot\delta^2/2}$$

Beweis:

$$\text{Prob}[X < (1-\delta)\cdot\mu] = \text{Prob}[-X > -(1-\delta)\mu]$$

$$= \text{Prob}\left[e^{-t\cdot X} > e^{-t(1-\delta)\cdot\mu}\right] \text{ für } t > 0$$

Analog zum Beweis von Chernoffs Schranke I erhalten wir

$$\text{Prob}[X < (1-\delta)\cdot\mu] < \frac{e^{(e^{-t}-1)\cdot\mu}}{e^{-t(1-\delta)\cdot\mu}}$$

Sei $t = \ln\left(\frac{1}{1-\delta}\right) = -\ln(1-\delta) \geq 0$

$$\rightarrow \text{Prob}[X < (1-\delta)\cdot\mu] < \left[\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right]^{\mu}$$

Für $\delta \in (0,1)$ folgt mit der Potenzreihenentwicklung von $\ln(1-\delta)$:

$$(1-\delta)^{(1-\delta)} > e^{-\delta+\frac{1}{2}\cdot\delta^2}$$

$\longrightarrow \text{Prob}[X < (1-\delta)\cdot\mu] <$

$$\left[\frac{e^{-\delta}}{e^{-\delta+1/2\cdot\delta^2}}\right]^{\mu} = e^{-\frac{1}{2}\cdot\delta^2\cdot\mu} \qquad \square$$

Anwendung 1: Routing im Hypercube

Sei G ein Netzwerk (= gerichteter Graph) mit N Knoten $1, 2, \ldots, N$ (= Prozessoren).

Entlang einer Kante kann in jedem Zeitschritt höchstens ein Packet geschickt werden.

- Netzwerk arbeitet synchron:
  - Zu jedem Zeitpunkt $t \in \mathbb{N}$ kann ein Prozessor i zu jedem seiner Nachbarn maximal ein Packet schicken.
  - Packete kommen sofort bei den Nachbarn an.
  - Nächster Zeitpunkt $= t+1$.

- Permutations-Routing-Problem

- Am Anfang: Jeder Prozessor $i \in \{1, \ldots, N\}$ besitzt ein Packet v_i. Sei $d(i) \in \{1, \ldots, N\}$ der Zielknoten des Packets v_i. Hierbei ist d eine Permutation.
- Ziel: Jedes Packet v_i soll möglichst schnell beim Zielprozessor $d(i)$ ankommen

Frage: Wieviele Schritte sind hierfür notwendig?

- Ein Routing-Algorithmus spezifiziert
  (1) für jedes Packet v_i eine Route, d.h. einen Pfad im Netzwerk G vom Startknoten i zum Zielknoten $d(i)$, entlang der das Packet v_i geschickt wird.
  (2) welches Packet entlang einer Kante e aus G zum Zeitpunkt t geschickt wird, falls mehrere Packete zum Zeitpunkt t entlang e laufen wollen.

- Ein Routing-Algorithmus ist Starr (engl: oblivious), falls die Route für Packet v_i nur vom $d(i)$ aber nicht vom $d(j)$ für $j \neq i$ abhängt.

  ↳ Route für v_i kann lokal von Prozessor i berechnet werden.

Satz (ohne Beweis): Sei A ein beliebiger deterministischer starrer Routing-Algorithmus für ein Netzwerk G mit N Knoten, wobei jeder Knoten in G Ausgangsgrad δ hat. Dann existiert eine Instanz des Permutations-Routing-Problems (d.h. eine Permutation $d: \{1, \ldots, N\} \to \{1, \ldots, N\}$ mit $d(i) =$ Zielknoten von v_i), für die A mindestens $\Omega(\sqrt{N/\delta})$ viele Schritte benötigt bis jedes Packet v_i am Zielknoten $d(i)$ ist.

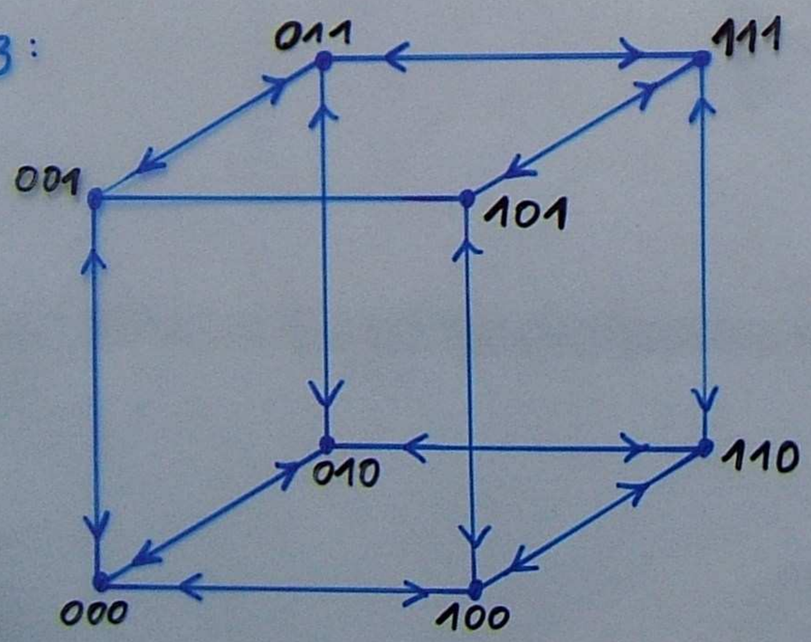
Wir betrachten im folgenden ein spezielles Netzwerk: Den

Hypercube H_n der Dimension n.

$H_n = (\{0,1\}^n, E)$ mit

$E = \{(u,v) \mid u,v \in \{0,1\}^n$, u und v unterscheiden sich an genau einer Position$\}$

H_3:



Bachte: H_n hat 2^n Knoten und jeder Knoten hat Ausgangsgrad n.

Bit-Fixing-Route vom Prozessor $i \in \{0,1\}^n$ zum Prozessor $d(i) \in \{0,1\}^n$:

Sei $i = u_0 i_1 u_1 i_2 \cdots u_{k-1} i_k u_k$

$d(i) = u_0 j_1 u_1 j_2 \cdots u_{k-1} j_k u_k$

mit $u_0, u_1, \ldots, u_k \in \{0,1\}^*$,

$i_1, j_1, \ldots, i_k, j_k \in \{0,1\}$ und

$i_\ell \neq j_\ell$ für $1 \leq \ell \leq k$

Die Bit-Fixing Route von i nach $d(i)$ ist:

$i = u_0 i_1 u_1 i_2 \cdots u_{k-1} i_k u_k \longrightarrow$

$u_0 j_1 u_1 i_2 \cdots u_{k-1} i_k u_k \longrightarrow$

$u_0 j_1 u_1 j_2 \cdots u_{k-1} i_k u_k \longrightarrow \cdots$

$u_0 j_1 u_1 j_2 \cdots u_{k-1} j_k u_k = d(i)$

Beispiel:

$0000 \rightarrow 1000 \rightarrow 1010 \rightarrow \underbrace{1011}_{d(i)}$

i

Wir betrachten den randomisierten Routing-Algorithmus Rand-Routing:

Phase 1: Prozessor i wählt unabhängig von den anderen Prozessoren $j \neq i$ einen Zwischenknoten $\sigma(i) \in \{0,1\}^n$ zufällig und gleichverteilt aus.
Paket v_i reist von i nach $\sigma(i)$ entlang der Bit-Fixing-Route.

Phase 2: Paket v_i reist von $\sigma(i)$ zum Zielknoten $d(i)$ entlang der Bit-Fixing-Route.

- Wollen zu einem Zeitpunkt t mehrere Pakete entlang einer Kante $j \longrightarrow j'$ reisen, so wählt j ein beliebiges Paket aus und sendet es nach j'
- Beachte:
  - Die Funktion $\sigma: \{0,1\}^n \rightarrow \{0,1\}^n$ ist i.A. keine Permutation.
  - Rand-Routing kann als ein randomisierter statischer Routing-Algorithmus angesehen werden: Prozessor i wählt die Route für Paket v_i unabhängig von den anderen Prozessoren aus.

Wir analysieren zunächst Phase 1.

Anzahl der Schritte, bis Packet v_i bei $\sigma(i)$ ankommt =
Länge der Route von i nach $\sigma(i)\} \leq n$
+ Anzahl der Schritte, bei denen Packet v_i in einem Zwischenknoten warten muß.

- Sei p_i die Bit-Fixing-Route von i nach $\sigma(i)$.

$$p_i = (\underset{\| \; i}{p_{i,0}}, p_{i,1}, \dots, \underset{\| \; \sigma(i)}{p_{i,i_k}})$$

Lemma 1 Wenn sich zwei Routen p_i und p_j $(i \neq j)$ irgendwann trennen, dann treffen sie sich später nie wieder.

Formal: Gelte $p_{i,a} = p_{j,b}$ aber $p_{i,a+1} \neq p_{j,b+1}$ $(i \neq j, 0 \leq a < i_k, 0 \leq b < j_k$.

$\longrightarrow p_{i,c} \neq p_{j,d}$ falls $c > a$, $d > b$.

Beweis:

Sei $p_{i,a+1} = u_1 k u_2 \neq v_1 \ell v_2 = p_{j,b+1}$

$p_{i,a} = u_1(1-k)u_2 = v_1(1-\ell)v_2 = p_{j,b}$

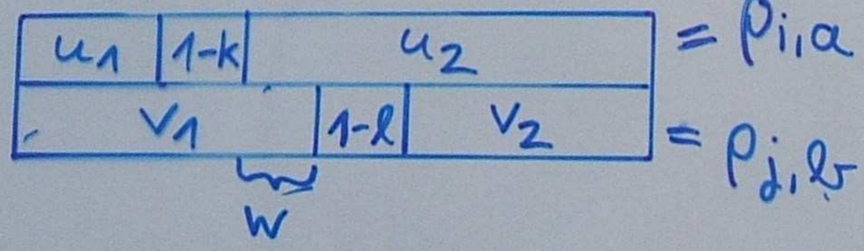
für $u_1, u_2, v_1, v_2 \in \{0,1\}^*$, $k, \ell \in \{0,1\}$

Sei o.B.d.A u_1 ein Präfix von v_1.
Aus $u_1 = v_1$ würde $k = \ell$ und $u_2 = v_2$ d.h. $p_{i,a+1} = p_{j,b+1}$ folgen. ↯

$\longrightarrow |u_1| < |v_1|$

$\longrightarrow v_1 = u_1(1-k)w$ und

$u_2 = w(1-l)v_2$



p_i und p_j sind Bit-Fixing-Routen

$\longrightarrow$ Jeder Wort $p_{i,c}$ $(c > a)$ beginnt mit $u_1 k$

Jeder Wort $p_{j,d}$ $(d > b)$ beginnt mit $v_1 \equiv u_1 (1-k) w$

$\rightarrow p_{i,c} \neq p_{j,d}$ falls $c > a$ und $d > b$

□

Sei im folgenden e_a die Kante

$p_{i,a-1} \rightarrow p_{i,a}$

<u>Lemma 2</u> Sei S die Menge aller Pakete v_j mit $j \neq i$, so dass die Route p_j eine der Kanten e_a durchläuft:

$$S = \{ v_j \mid j \neq i, \exists a, b : (p_{i,a-1}, p_{i,a}) = (p_{j,b-1}, p_{j,b}) \}$$

Dann ist die Anzahl der Schritte, bei denen Paket v_i an einem der Zwischenknoten $p_{i,a}$ $(0 \leq a < i_k)$ warten muß, höchstens $|S|$.

<u>Beweis</u>:

- Paket $v \in S$ <u>verlässt Route</u> p_i zum Zeitpunkt t :$\Longleftrightarrow$ v durchläuft zum Zeitpunkt t zum letzten mal eine der Kanten e_a von p_i.

- Ist Paket $v \in S \cup \{v_i\}$ zum Zeitpunkt t bereit, die Kante e_a zu durchlaufen, so ist $t-a$ die Verzögerung vom Paket v zum Zeitpunkt t.

→ Anzahl von Schritten bei denen Paket v_i an Zwischenknoten warten muß =
Verzögerung von v_i zu dem Zeitpunkt, wo v_i die letzte Kante e_{i_k} von p_i durchläuft.

- Angenommen, zum Zeitpunkt t erhöht sich die Verzögerung vom Packet v_i von l auf $l+1$.

→ Es gibt Packet $v \in S$, welches zum Zeitpunkt t die gleiche Kante wie v_i durchlaufen möchte.

(sonst hätte v_i zum Zeitpunkt t eine Kante durchlaufen und die Verzögerung von v_i wäre gleich geblieben)

→ $\exists v \in S$: Verzögerung von v erreicht l

Sei $t' = \max\{t'' \mid \exists v \in S$: Verzögerung von v zum Zeitpunkt $t'' = l\}$

→ $t' \geq t$

Sei $v \in S$ ein Paket, welches zum Zeitpunkt t' bereit ist, die Kante e_a zu durchlaufen und $l = t' - a$.

→ Es gibt ein Paket v' welches zum Zeitpunkt t' die Kante e_a durchläuft.

Es gilt $v' \neq v_i$, denn v_i durchläuft

keine Kante zum Zeitpunkt t, und zu einem Zeitpunkt $t'' > t$ hat v_i schon eine Verzögerung $> l$.

$\longrightarrow v' \in S$.

- v' muß Route p_i zum Zeitpunkt t' verlassen, denn sonst wäre v' zum Zeitpunkt $t'+1$ bereit die Kante e_{a+1} zu durchlaufen. $\longrightarrow$ Verzögerung von v' zum Zeitpunkt $t'+1 = t'+1-(a+1) =$
$= t'-a = l$ $\lightning$ Widerspruch zur Wahl von t'.

Wir haben nun folgende Implikation bewiesen.

Erhöht sich die Verzögerung von Paket v_i um 1, so verlässt (zu einem evtl. späteren Zeitpunkt) ein Paket aus S die Route p_i

Lemma 1 $\longrightarrow$ Jedes Paket aus S kann nur einmal die Route p_i verlassen.

$\longrightarrow$ Verzögerung von Paket v_i kann sich nur höchstens $|S|$ mal um 1 erhöhen.

$\longrightarrow$ Anzahl der Schritte, bei denen Paket v_i an einem Zwischenknoten warten muß $\leq |S|$. □

- Definiere Zufallsvariable

$$H_{i,j} = \begin{cases} 1 & \text{falls Route } p_i \text{ und } p_j \\ & \text{eine gemeinsame Kante haben} \\ 0 & \text{sonst} \end{cases}$$

- Lemma 2 $\longrightarrow$

Schritte, bei denen Packet p_i an einem Zwischenknoten warten muß

$$\leq \sum_{j \neq i} H_{i,j}$$

- Für eine Kante e des Hypercube definiere die Zufallsvariable

$$T_e = |\{i \mid \text{Route } p_i \text{ durchläuft } e\}|$$

- Für $p_i = (e_1, \dots, e_k)$ gilt:

$$\sum_{j \neq i} H_{i,j} \leq \sum_{i=1}^{k} T(e_i)$$

$\longrightarrow$

$$E\left[\sum_{j \neq i} H_{i,j}\right] \leq E\left[\sum_{i=1}^{k} T(e_i)\right] \quad (i \text{ fest})$$

$$= \sum_{i=1}^{k} \overset{E[T(e)]}{E[T(e_i)]} = k \cdot \overset{E[T(e)]}{T(e)} \leq m \cdot T(e)$$

(für beliebige Kante e)

- Für alle $j \in \{0,1\}^n$ gilt:

$$E[\text{Länge der Route } p_j] = \frac{n}{2}$$

$$\sum_{\substack{e \text{ Kante} \\ \text{vom } H_m}} E[T(e)] = m \cdot 2^m \cdot \overbrace{T(e)}^{E[T(e)]}$$

$$\| $$

$$E\Big[\sum_{\substack{e \text{ Kante} \\ \text{vom } H_m}} T(e)\Big] =$$

$$= E\Big[\sum_{j \in \{0,1\}^m} \text{Länge der Route } p_j\Big] =$$

$$= \sum_{j \in \{0,1\}^m} \overbrace{E[\text{Länge der Route } p_j]}^{\frac{m}{2}} =$$

$$\frac{m}{2} \cdot 2^m$$

$$\longrightarrow E[T(e)] = \frac{1}{2}$$

$$\longrightarrow E\Big[\sum_{j \neq i} H_{i,j}\Big] \leq \frac{m}{2}$$

Sei $\mu = E\Big[\sum_{j \neq i} H_{i,j}\Big] \leq \frac{m}{2}$

$$\rightarrow \text{Prob}\Big[\sum_{j \neq i} H_{i,j} > \overbrace{6m}^{(1+\delta)}\Big]$$

$$= \text{Prob}\Big[\sum_{j \neq i} H_{i,j} > \frac{6 \cdot m}{\mu} \cdot \mu\Big]$$

Sei $\delta = \frac{6m}{\mu} - 1 \geq^{5} 11 > 2 \cdot e - 1$

Chernoff (Korollar aus Chernoffs Schranke I) $\longrightarrow$

$$\text{Prob}\Big[\sum_{j \neq i} H_{i,j} > 6m\Big] < 2^{-(1+\delta)\cdot\mu}$$

$$= 2^{-6m}$$

Prob[Paket i muß in Phase 1 bei mehr als $^{3}6n$ Schritten warten]

$< 2^{-^{3}6n}$

Es gibt 2^n Packete $\longrightarrow$

Prob[ein Paket muß in Phase 1 bei mehr als $^{3}6n$ Schritten warten]

$< 2^n \cdot 2^{-^{3}6n} = 2^{-^{2}5n}$

Da die Länge jeder Route höchstens n ist, folgt:

Prob[Jedes Paket i erreicht in Phase 1 nach höchstens $^{4}7n$ Schritten das Zwischenziel $\sigma(i)$] $\geq 1 - 2^{-^{2}5n}$

Phase 2 ist nichts anderes als Phase 1 rückwärts betrachtet.
(hier wichtig: d ist Permutation)

$\longrightarrow$ Die gleiche Analyse wie für Phase 1 zeigt:

Prob[Jedes Paket i erreicht in Phase 2 sein Ziel d(i) in höchstens $^{4}7n$ Schritten]

$\geq 1 - 2^{-^{2}5n}$

<u>Vorsicht</u>: Pakete, die noch in Phase 1 sind, können Pakete aus Phase 2 verzögern und umgekehrt.

<u>Lösung</u>: Die Wahrscheinlichkeit hierfür kann auf $2^{-^{2}5n}$ abgesenkt

werden, in dem jedes Packet i am Zwischenknoten $G(i)$ so lange warten muß bevor es in Phase 2 geht, bis mindestens $7n$ [4] Schritte vergangen sind.

→ $Prob[$ jedes Packet i erreicht Ziel $d(i)$ in $\leq 14n$ [8] Schritten $]$

$\geq 1 - 2 \cdot 2^{-5n}$ [2]

$= 1 - 2^{-5n+1} \geq 1 - 2^{-n}$