Big Data Übungsblatt 08

Anton Bulat, Josephine Geiger, Julia Siekiera

December 9, 2017

Aufgabe 1: Kommunikationskosten

Bestimmen Sie die Kommunikationskosten der folgenden Probleme in Abhängigkeit der genannten Eingabegrößen:

a)

Schnitt zweier Relationen $R \cap S$ mit r bzw. s Tupeln (Folie 38, Vorlesung 5):
Die Eingabe in den Mapper besteht aus allen Tupeln aus beiden Relationen, also sind die Kommunikationskosten hier in $\mathcal{O}(r+s)$. Da der Mapper nur die Identität ist, ist die Eingabe in den Reducer genauso groß. Also sind die Gesamtkommunikationskosten dieses Problems in $\mathcal{O}(2r+2s) = \mathcal{O}(r+s)$.

b)

Gruppenbasierter Similarity-Join mit n Bildern und Gruppengröße h (Folie 39, Vorlesung 7):
Die Eingabe in den Mapper besteht aus Tupeln (i, P_i) mit dem Bildindex und dem Bild. Bei n Bildern sind diese Kommunikationskosten also in $\mathcal{O}(n)$. Die Eingabe in den Reducer ist höher. Es werden die n Bilder an jeweils $g-1$ Reducer geschickt, also liegen diese Kommunikationskosten in $\mathcal{O}(n \times \frac{n}{h})$. Die Gesamtkommunikationskosten dieses Problems liegen somit in $\mathcal{O}(\frac{n^2}{h} + n)$.

Aufgabe 2: Graphische Modelle

a)

Grundsätzlich lassen sich anhand eines graphischen Modells Aussagen treffen über

- die minimal mögliche Reducergröße q:

  $$q \geq \max_{n \in B} deg(n).$$

  Das heißt, man braucht mindestens so viele Eingaben in einem Reducer wie die höchste Anzahl an Eingangskanten in einen Ausgabeknoten.

- die maximal benötigte Replikationsrate r:

$$r \leq \sum_{n \in A} deg(n) = \sum_{n \in B} deg(n).$$

  Das heißt, man braucht höchstens so viele Replikationen einer einzelnen Eingabe wie Kanten insgesamt. (Noch öfter braucht eine Eingabe nicht geschickt zu werden.)

b)

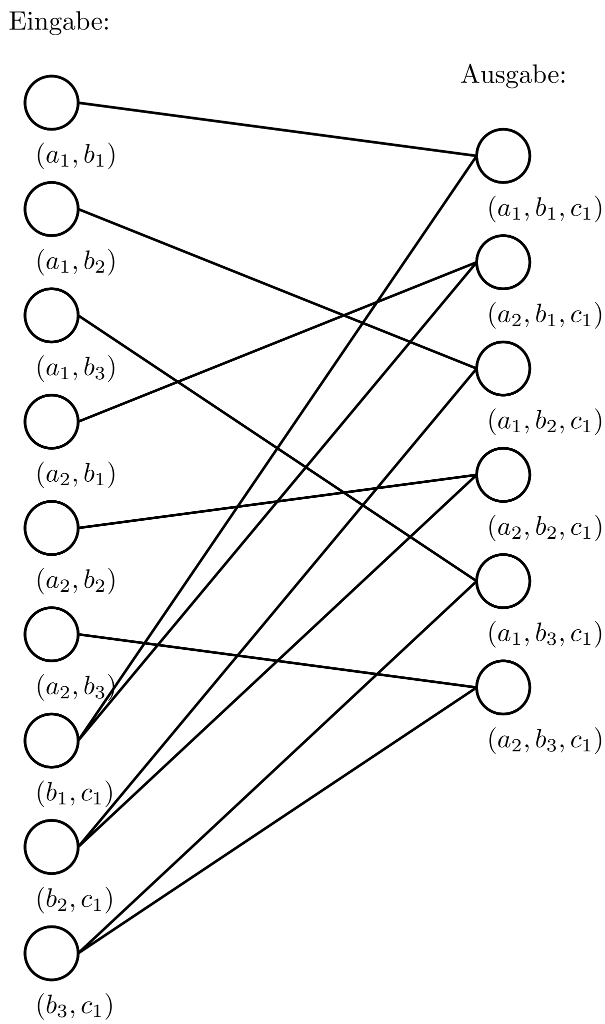
Natürlicher Join $R(A, B) \bowtie S(B, C)$ mit a möglichen Werten für A, b möglichen Werten für B und c möglichen Werten für C:

i)

Das zugehörige graphische Modell besitzt $a*b+b*c$ Eingabeknoten und $a*b*c$ Ausgabeknoten.

ii)

Modell für $a = 2$, $b = 3$ und $c = 1$:



iii)

Schranken für q und r in Abhängigkeit von a, b und c:

$$q \geq \max_{n \in B} deg(n) \Rightarrow q \geq 2.$$

Die untere Schranke für q hängt hier nicht von a, b und c ab, sondern ist hier fest ≥ 2, weil zwei Relationen gejoint werden und deshalb zwei Eingabetupel für ein mögliches Ausgabetupel benötigt werden.

$$r \leq \sum_{n \in B} deg(n) \Rightarrow r \leq 2 * a * b * c.$$

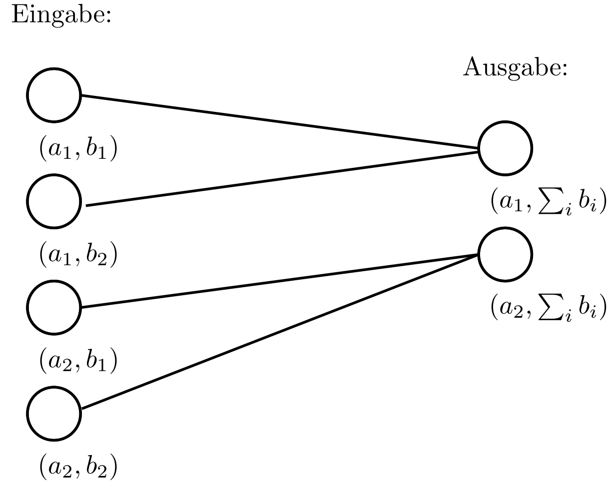
c)

Gruppierung $\gamma_{A,SUM(B)}R(A,B)$ mit a möglichen Werten für A und b möglichen Werten für B:

i)

Das zugehörige graphische Modell besitzt $a*b$ Eingabeknoten (gleich der Anzahl an Tupeln in der Relation R) und a Ausgabeknoten, da für jeden Wert von A eine Summe berechnet wird.

ii)



iii)

Schranken für q und r in Abhängigkeit von a und b:

$$q \geq \max_{n \in B} deg(n) \Rightarrow q \geq b.$$

Für jeden Wert von A gibt es bis zu b Summanden, wofür jeweils ein Tupel an den Reducer geschickt werden muss.

$$r \leq \sum_{n \in B} deg(n) \Rightarrow r \leq a * b.$$

Aufgabe 3: Unterschranken an Replikationsraten

a)

Der Reducer vergleicht paarweise die Tupel aus R und S, womit bei q Eingabewerten maximal $\binom{q}{2} \approx q^2/2$ Ausgabewerte überdeckt werden. Damit ist $g(q) \approx$

$q^2/2$ für $q \geq 2$. Für $q < 2$ ist $g(q) = 0$.
Falls man davon ausgeht, dass der Algorithmus zur Berechnung des Schnitts dem Algorithmus auf der Folie 38 der Vorlesung 5 entspricht, wird dem Reducer zu einem beliebigen Tupel, das in R oder S enthalten ist, ein Key/Value-Paar mit genau zwei Tupeln als Value übergeben, da R und S die gleichen Tupel enthalten. Somit ist unser q immer gleich 2 und ein Reducer gibt genau einen Ausgabewert aus. Somit ist $g(q) = 1$.

b)

Wenn $A \in \mathbb{R}^{m \times n}$ und $\vec{x}, \vec{b} \in \mathbb{R}^{n \times 1}$, dann muss $m = n$ gelten, damit die Gleichung $A\vec{x}+\vec{b}$ eine gültige Lösung $\vec{y}$ mit $\vec{y} \in \mathbb{R}^{n \times 1}$ hat (Voraussetzung gleiche Dimension bei Vektoraddition). Der Reducer berechnet einzeln die Werte y_i, womit bei q Eingabewerten maximal $\binom{q}{1} = q$ Ausgabewerte überdeckt werden. Damit ist $g(q) = q$. Bei $m \neq n$ ist $g(q) = 0$, da es keine gültige Lösung gibt.

c)

- Schritt 1:
  Laut Aufgabenstellung $g(q) = \frac{\sqrt{2}}{3} q^{\frac{3}{2}}$.

- Schritt 2:
  Als Ausgabe werden Tripel von Kanten erwartet, die jeweils eine 3-Clique bilden. In einem ungerichteten Graph mit n Knoten kann es maximal $\binom{n}{3}$ 3-Cliquen für $n > 2$ geben (wenn der Graph vollständig ist). Also ist die gesamte Anzahl der Ausgaben nicht gröer als $\binom{n}{3}$ bzw. man kann zu jeder 3er Kombination der Knoten angeben, ob es sich um eine Clique handelt oder nicht. Damit wäre die Gesamtzahl der vom Problem generierten Ausgaben $m = \binom{n}{3} = \frac{n*(n-1)*(n-2)}{6} \approx \frac{n^3}{6}$.

- Schritt 3:
  Es gilt $\sum_{i=1}^{k} g(q_i) \geq m$ mit $g(q_i) = \frac{\sqrt{2}}{3} q_i^{\frac{3}{2}}$ und $m \approx \frac{n^3}{6}$, also $\Rightarrow \sum_{i=1}^{k} \frac{\sqrt{2}}{3} q_i^{\frac{3}{2}} \geq \frac{n^3}{6} \Leftrightarrow \sum_{i=1}^{k} 2 * \sqrt{2} q_i^{\frac{3}{2}} \geq n^3 \Leftrightarrow \sum_{i=1}^{k} 2 * \sqrt{2 * q_i^3} \geq n^3$.

- Schritt 4:
  Da $q \geq q_i$ bleibt die Ungleichung $\sum_{i=1}^{k} 2 * \sqrt{2 * q_i^2 * q} \geq n^3$ erfüllt.

- Schritt 5:
  Etwas umgeformt ergibt sich die untere Schranke für r: $\sum_{i=1}^{k} q_i \geq \frac{n^3}{2*\sqrt{2*q}} \Leftrightarrow \frac{1}{n} * \sum_{i=1}^{k} q_i = r \geq \frac{n^2}{2*\sqrt{2*q}}$.
  Kommentar: Man kann in dem letzten Term das n^2 mit $(n-1)*(n-2)$ ersetzen, falls man beim Schritt 2 nicht approximieren möchte, sondern den genauen Ausdruck nimmt.

d)

- Schritt 1:
  Laut Aufgabenstellung $g(q) = \frac{q}{2} * \log_2(q)$.

- Schritt 2:
  $|M| = n$ ist die Anzahl der Elemente in der gegebenen Menge M und b ist die Länge der in M vorkommenden Bitstrings. Ein Bitstring-Paar gehört zur Ausgabemenge, wenn die zwei Elemente sich genau um ein Bit unterscheiden. Deshalb hat jeder Bitstring potenziell b mögliche Elemente in M, mit denen er ein Paar bilden kann, das zur Ausgabemenge gehört. Es ist der Fall, wenn sich genau ein Zeichen des Bitstrings von "0" auf "1" oder umgekehrt ändert. Mit n vielen Elementen in M ergibt sich die Gesamtzahl der vom Problem generierten Ausgaben $m = \binom{n}{b} = \frac{n!}{b!*(n-b)!}$.
  Es gilt auch $2 \leq n \leq b^2$, da es maximal b^2 Elemente geben kann, ohne dass ein Element in der Menge doppelt vorkommt und es mindestens zwei Elemente in M vorhanden sein müssen, um eine gültige Ausgabe zu generieren.

- Schritt 3:
  Es gilt $\sum_{i=1}^{k} g(q_i) \geq m$ mit $g(q) = \frac{q}{2} * \log_2(q)$ und $m = \frac{n!}{b!*(n-b)!}$, also
  $\Rightarrow \sum_{i=1}^{k} \frac{q_i}{2} * \log_2(q_i) \geq \frac{n!}{b!*(n-b)!} \Leftrightarrow \sum_{i=1}^{k} q_i * \log_2(q_i) \geq \frac{2*n!}{b!*(n-b)!}$.

- Schritt 4:
  Da $q \geq q_i$ und $\log_2(q)$ monoton wachsend ist, bleibt die Ungleichung $\sum_{i=1}^{k} q_i * \log_2(q) \geq \frac{2*n!}{b!*(n-b)!}$ erfüllt.

- Schritt 5:
  Etwas umgeformt ergibt sich die untere Schranke für r: $\sum_{i=1}^{k} q_i \geq \frac{2*n*(n-1)!}{b!*(n-b)!*\log_2(q)} \Leftrightarrow$ $\frac{1}{n} * \sum_{i=1}^{k} q_i = r \geq \frac{2*(n-1)!}{b!*(n-b)!*\log_2(q)}$.