On Finding Similar Items in a Stream of Transactions

Andrea Campagna, Rasmus Pagh

IT University of Copenhagen

December 14, 2010
The problem

$[n] = \{1, \ldots, n\}$
The problem

$U = \{1, \ldots, n\}$
The problem

$[n] = \{1, \ldots, n\}$
The problem

\[ \mathcal{U} \]

\[ \{1, \ldots, n\} \]
The problem

\[ [n] = \{1, \ldots, n\} \]

Andrea Campagna, Rasmus Pagh
On finding similar items in a stream of transactions
The problem

\[ [n] = \{1, \ldots, n\} \]
The problem

\[ U = \{1, \ldots, n\} \]

Transaction \( T \)

\[ S_2 := \{ \ldots \} \]
The problem

Transaction $T$:

$S_2 := \{ \text{golden yellow}, \text{blue} \}$

$[n] = \{1, \ldots, n\}$

$U$
The problem

\[ \mathcal{U} \]

\[ [n] = \{1, \ldots, n\} \]

Transaction \( T \)

\[ S_2 := \{ \] Similar pairs

\[ \} \]
The problem

\[ U = \{1, \ldots, n\} \]

Transaction \( T \)

\[ S_2 := \{ \text{similar pairs} \} \]

Andrea Campagna, Rasmus Pagh
On finding similar items in a stream of transactions
The problem
The problem

Andrea Campagna, Rasmus Pagh
On finding similar items in a stream of transactions
The problem

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The problem

Similarity?

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The problem

\[ \text{Measure} ((i, j) f |S_i|, |S_j|) \]

- Cosine
- Dice
- All confidence
- Overlap coef
The problem

| Measure              | $s(i, j)$                  | $f(|S_i|, |S_j|)$ |
|----------------------|---------------------------|-----------------|
| **Cosine**           | $\frac{|S_i \cap S_j|}{\sqrt{|S_i||S_j|}}$ | $\frac{1}{\sqrt{|S_i| \cdot |S_j|}}$ |
| **Dice**             | $\frac{|S_i \cap S_j|}{|S_i| + |S_j|}$    | $\frac{1}{(|S_i| + |S_j|)}$ |
| **All_confidence**   | $\frac{|S_i \cap S_j|}{\max(|S_i|, |S_j|)}$ | $\frac{1}{\max(|S_i|, |S_j|)}$ |
| **Overlap_coef**     | $\frac{|S_i \cap S_j|}{\min(|S_i|, |S_j|)}$ | $\frac{1}{\min(|S_i|, |S_j|)}$ |

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The problem

\[
\cos(1, 3) = \frac{3}{\sqrt{12}} \approx 86\%
\]

\[
\cos(1, 6) = \frac{2}{\sqrt{8}} \approx 71\%
\]

\[
\cos(2, 6) = \frac{2}{\sqrt{4}} = 100\%
\]
The problem

\[ \cos(1, 3) = \frac{3}{\sqrt{12}} \approx 86\% \]

\[ \cos(1, 6) = \frac{2}{\sqrt{8}} \approx 71\% \]

\[ \cos(2, 6) = \frac{2}{\sqrt{4}} = 100\% \]
The problem

\[ \cos(1, 3) = \frac{3}{\sqrt{12}} \approx 86\% \]

\[ \cos(1, 6) = \frac{2}{\sqrt{8}} \approx 71\% \]
The problem

\[
\cos(1, 3) = \frac{3}{\sqrt{12}} \approx 86\%
\]

\[
\cos(1, 6) = \frac{2}{\sqrt{8}} \approx 71\%
\]

\[
\cos(2, 6) = \frac{2}{\sqrt{4}} = 100\%
\]
The problem

\[ \cos(1, 3) = \frac{3}{\sqrt{12}} \approx 86\% \]

\[ \cos(1, 6) = \frac{2}{\sqrt{8}} \approx 71\% \]

\[ \cos(2, 6) = \frac{2}{\sqrt{4}} = 100\%!!! \]
The algorithm

Pair sampling

From similarity to frequency

Sample count

Frequency filtering

Mod(BiSam)

Mod(Demaine et al.)

\{x_p, x_q\}

\{x_i, x_j\}

...
The algorithm

Pair sampling

From similarity to frequency

Sample count

Frequency filtering

\{x_p, x_q\}

\{x_i, x_j\}

\ldots

Mod(BiSam)

Mod(Demaine et al.)

Random order

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The algorithm - From similarity to frequency

...
The algorithm - From similarity to frequency

$S_k$

...
The algorithm - From similarity to frequency

\[ S_k \]

\[ S_{2k} \]
The algorithm - From similarity to frequency

\[ S_k \quad S_{2k} \quad S_{4k} \]

Andrea Campagna, Rasmus Pagh
On finding similar items in a stream of transactions
The algorithm - From similarity to frequency

\[ S_k \]
\[ S_{2k} \]
\[ S_{4k} \]
\[ S_{8k} \]
The algorithm

- From similarity to frequency

\[ S_k \]
\[ S_{2k} \]
\[ S_{4k} \]
\[ S_{8k} \]...

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The algorithm - From similarity to frequency

\[ \cos(i, j) = \frac{|S_i \cap S_j|}{\sqrt{|S_i| \cdot |S_j|}} = |S_i \cap S_j| \cdot \frac{1}{\sqrt{|S_i| \cdot |S_j|}} \]
The algorithm - From similarity to frequency

**BiSam**

\[
\Pr\{i, j\ \text{sampled}\} = \frac{1}{\tau \sqrt{|S_i| \cdot |S_j|}};
\]

\[
\text{rand}_{T_k} < \tau \frac{1}{\sqrt{|S_i| \cdot |S_j|}};
\]

Row by row.

\[
T_k[1] \quad T_k[2] \quad \ldots
\]

\[
T_k
\]

\[
\phi
\]

Independent samples \( O(\log(mn)) \)
The algorithm - From similarity to frequency

Mod(BiSam)

- \( \Pr[\{i, j\} \text{ sampled}] = \frac{1}{\tau \sqrt{|S_i| \cdot |S_j|}}; \)
- \( \text{rand}_{T_k} < \frac{1}{\tau \sqrt{|S_i| \cdot |S_j|}}; \)
- Row by row.

\( T_k \)

\( T_k[1] \)

\( T_k[2] \)

\( \ldots \)

\( \varphi \)
The algorithm - From similarity to frequency

Mod(BiSam)

- \[ \Pr[\{i, j\} \text{ sampled}] = \frac{1}{\tau \sqrt{|S_i| \cdot |S_j|}}; \]
- \[ \text{rand}_{T_k} < \tau \frac{1}{\sqrt{|S_i| \cdot |S_j|}}; \]
- Row by row.

\[ T_k \]

\[ T_k[1] \]

\[ T_k[2] \]

...
The algorithm - From similarity to frequency

Mod(BiSam)

- \( \Pr[\{i, j\} \text{ sampled}] = \frac{1}{\tau \sqrt{|S_i| \cdot |S_j|}}; \)
- \( \text{rand}_{T_k} < \tau \frac{1}{\sqrt{|S_i| \cdot |S_j|}}; \)
- Row by row.

\[ O(|T_k| \log(mn)) \]
The algorithm
- Frequency filtering

\[
\text{Pr} = \sum_{i=1}^{p_{s/2}} \text{ek}_{i} = p_{i} \text{count} + \ldots
\]
The algorithm

- Frequency filtering

\[ \text{Pr} = \frac{s}{2t} \]

Andrea Campagna, Rasmus Pagh

On finding similar items in a stream of transactions
The algorithm - Frequency filtering

\[ \Pr = \frac{s}{2} \]

\[ e_k = \sum_{i} p_i. \]
The algorithm

- Frequency filtering
The algorithm - Frequency filtering

\[ Pr = \frac{s}{2t} \]

Andrea Campagna, Rasmus Pagh
On finding similar items in a stream of transactions
The algorithm - Frequency filtering

\[
\frac{s}{2} p_1 \ldots \frac{s}{2} p_i \ldots \frac{s}{2} p_t
\]

\[
Pr = \frac{s}{2t}
\]

Andrea Campagna, Rasmus Pagh
The algorithm - Frequency filtering

\[ p_1, p_{s/2} \rightarrow p_t \]

\[ Pr = s/2t \]

\[ p_1 \]
\[ \vdots \]
\[ p_i \]
\[ \vdots \]
\[ p_{s/2} \]

\[ e_k = p_i \]

\[ p_1 \]
\[ \vdots \]
\[ p_i \]
\[ \vdots \]
\[ p_{s/2} \]

\[ p_{s/2} \rightarrow p_i \]

\[ p_{s/2} \rightarrow p_i \]

\[ p_{s/2} \rightarrow p_i \]

\[ p_{s/2} \rightarrow p_i \]

\[ p_{s/2} \rightarrow p_i \]
Our results

- First approach to similarity in a streaming context;
- maintains and reports whp pairs with similarity over a certain threshold;
- $O(mb \log(nm))$ whp for processing prefixes of size $mb$;
- $O(n + s)$ space;
One more result - a space lower bound

$$\Omega\left(\min\left(m, n^k, \left(\frac{m}{\varphi}\right)^k\right)\right)$$
Open problems

- Experiments;
- Possibly using tools like MapReduce;
- Extension of the lower bound in order to take into account the random order of transactions.
Open problems

- Experiments;
- Possibly using tools like MapReduce;
- Extension of the lower bound in order to take into account the random order of transactions.

Siamo qualcosa che non resta, frasi vuote nella testa, ed il cuore di simboli pieno.