21 – November 2012

Scribe Notes – Sundararajan Sarangan

Last time we talked about how data streaming works.

A Quick Summary:

We have a 2-D gigantic table which is a binary table. (0s and 1s). We also have a Transaction ID Field. For every two columns we need to do a bitwise AND and we need to count the number of 1s in the result, so that we can figure out which 2 columns have high correlation.

ID Milk Cereal

The Wal-Mart example:

Our approach is to summarize every column into a 'sketch', a vector of 100 mean value registers MR_1, MR_2, … , MR_{100}

This is done by taking a pass on every column to produce a MR for each column.

$MR_i^{(m)} \leftarrow \text{Min} \{ H_i(\text{transaction.ID}) \mid \text{Transaction contains "milk"} \}$

MR for milk column is the min of hash values for Tx ID for all transactions which have milk in them.

$| A \cap B | = | A | + | B | - | A \cup B |$

Here, A is the set of transactions that contain milk, B is the set of transactions that contain cereal. We are trying to find out how many transactions have both.

Lets put all the MRs in a 10 X 11 Matrix.

Means

MR_1	….	MR_{10}	MR_{1-10}
.			
.			
MR_{91}	….	MR_{100}	
MR_{101}	….	MR_{110}	$MR_{101-110}$

ESTIMATORS: Median of Mean (This is the estimator we are going to use)

Min { $MR_1^{(m)}$, $MR_1^{(c)}$ } → MR_1'

Min { $MR_2^{(m)}$, $MR_2^{(c)}$ } → MR_2'

.

.

Min { $MR_{110}^{(m)}$, $MR_{110}^{(c)}$ } → MR_{110}'

Now we put all these primes into the matrix.

Means

MR_1	….	MR_{10}	MR_{1-10}
.			
.			
MR_{91}	….	MR_{100}	
MR_{101}	….	MR_{110}	$MR_{101-110}$

The median of the means gives us the final answer, α. The number of distinct elements is given by $(1/\alpha) - 1$.

Now, Going back to the networking context:

Another Algo To Count Number of Distinct Elements:

The above solution isn't good when we have to do it online. For example, if we have 10s of millions of flow traffic, we will have to update 100 registers (in 8ns or so). To overcome this, we make use of a faster algorithm called Bitmap Algorithm.

BITMAP ALGORITHM

Suppose you have a bitmap of 8 million bits. This will take up 1 MB of space.

To set this array:

Foreach (datum) {

A[H(datum)] = 1;

}

Where

H : [ID] $\rightarrow$ { 1, 2, … , N}

H is a hash function that maps ID to a uniform random number between 1 and N.

At the end of the data stream, we will have this bit array, from which we will calculate our estimate.

Let M be the actual number of distinct elements. We do not know M. We are going to estimate M as follows:

i. Pick an arbitrary bit 'i'.
ii. Let X_i be the value of the bit i after stream has passed.
iii. Since we picked i randomly, X_i is random too.

X_i = { 1, $1 - (1 - 1/N)^M$
{ 0, $(1 - 1/N)^M$

Probability of each slot missing a location is i = 1 – 1/N. So, probability of X_i being 0 is $(1 - 1/N)^M$

iv. Let X be the number of 1s in the array.

$X = {}^N\sum_{i=1} X_i$
Expectation of X, $E[X] = {}^N\sum_{i=1} E[X_i]$

v. We know that $E[X_i]$ being 1 is $(1 - 1/N)^M$
So, $E[X] = N(1 - 1/N)^M$

vi. Now, method of moments is defined as C = E[X]
Therefore, $C = N(1 - (1 - 1/N)^M)$

Method of moments: We put two quantities on two sides of an equation. One is observed, and the other is theoretically calculated.
'C' is the observed number of 1s.

Solving,

$$\alpha = C/N = (1 - 1/N)^M$$

So,

$$(1 - 1/N)^M = 1 - \alpha$$

Multiplying and dividing the powers by M,

$$e^{-1(M/N)} = 1 - \alpha$$

Taking ln on both sides,

$$-M/N = \ln(1 - \alpha)$$

$$\text{So, estimator} = -N\ln(1 - \alpha)$$

Where α is the percentage of 1s in the array.