Information Processing Letters 115 (2015) 336–342

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A unified scheme for generalizing cardinality estimators to sum aggregation ☆

Reuven Cohen *, Liran Katzir, Aviv Yehezkel

Department of Computer Science, Technion, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 14 February 2014
Received in revised form 25 June 2014
Accepted 14 October 2014
Available online 22 October 2014
Communicated by M. Yamashita

Keywords:
Algorithms
Statistical
Big data processing

ABSTRACT

Cardinality estimation algorithms receive a stream of elements that may appear in arbitrary order, with possible repetitions, and return the number of distinct elements. Such algorithms usually seek to minimize the required storage at the price of inaccuracy in their output. This paper shows how to generalize every cardinality estimation algorithm that relies on extreme order statistics (min/max sketches) to a weighted version, where each item is associated with a weight and the goal is to estimate the total sum of weights. The proposed unified scheme uses the unweighted estimator as a black-box, and manipulates the input using properties of the beta distribution.

1. Introduction

Consider a very long stream of elements $x_1, x_2, \ldots, x_s$ with repetitions. Finding the number n of distinct elements is a well-known problem with numerous applications. The elements might represent IP addresses of packets passing through a router [8,9,16], elements in a large database [12], motifs in a DNA sequence [10], or elements of RFID/sensor networks [17]. One can easily find the exact value of n in the following way. When a new element x_i is encountered, compare its value to every distinct (stored) value encountered so far. If the value of x_i has not been seen before, keep it in the storage as well. After all the elements are treated, count the number of stored elements. This simple approach does not scale if storage is limited, or if the computation performed for each element x_i should be minimized. In such a case, the following cardinality estimation problem should be solved:

Problem 1 *(The cardinality estimation problem).*
Instance: A stream of elements $x_1, x_2, \ldots, x_s$ with repetitions, and an integer m. Let n be the number of different elements, namely $n = |\{x_1, x_2, \ldots, x_s\}|$, and let these elements be $\{e_1, e_2, \ldots, e_n\}$.
Objective: Find an estimate $\widehat{n}$ of n using only m storage units, where $m \ll n$.

Several algorithms have been proposed for the cardinality estimation problem. See [2] for a theoretical overview and [16] for a practical overview with comparative simulation results. In this paper we study the following weighted generalization of the cardinality estimation problem:

Problem 2 *(The weighted cardinality estimation problem).*
Instance: A stream of weighted elements $x_1, x_2, \ldots, x_s$ with repetitions, and an integer m. Let n be the number of different elements, namely $n = |\{x_1, x_2, \ldots, x_s\}|$, and let these elements be $\{e_1, e_2, \ldots, e_n\}$. Finally, let w_j be the weight of e_j.
Objective: Find an estimate $\widehat{w}$ of $w = \sum_{j=1}^{n} w_j$ using only m storage units, where $m \ll n$.

An example of an instance for the cardinality estimation problem is the stream: a, b, a, c, d, b, d. For this

☆ The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement No. 610802.

* Corresponding author.
E-mail address: rcohen@cs.technion.ac.il (R. Cohen).

http://dx.doi.org/10.1016/j.ipl.2014.10.009
0020-0190/

instance, $n = |\{a, b, c, d\}| = 4$. An example of an instance for the weighted problem is: $a(3)$, $b(4)$, $a(3)$, $c(2)$, $d(3)$, $b(4)$, $d(3)$. For this instance, $e_1 = a$, $e_2 = b$, $e_3 = c$, $e_4 = d$, $w_1 = 3$, $w_2 = 4$, $w_3 = 2$, $w_4 = 3$ and $\sum w_j = 12$.

As an application example, $x_1, x_2, \ldots, x_s$ could be IP packets received by a server. Each packet belongs to one of n IP flows $e_1, e_2, \ldots, e_n$. The weight w_j can be the load imposed by flow e_j on the server. Thus, $\sum_{j=1}^{n} w_j$ represents the total load imposed on the server by all the flows to which packets $x_1, x_2, \ldots, x_s$ belong.

The main contribution of this paper is a unified scheme for generalizing any extreme order statistics estimator for the unweighted cardinality estimation problem to an estimator for the weighted cardinality estimation problem. **This scheme can be used for obtaining known estimators and new estimators in a generic way.** In particular, we show in Section 6 that:

- The new scheme can be used to extend the HyperLogLog algorithm [6], originally developed for the unweighted problem, to solve the weighted problem. The extended algorithm offers the best performance, in terms of statistical accuracy and memory storage, among all the other known algorithms for the weighted problem.
- The new scheme can be used to extend the "data sketching with Bernoulli random variables" estimator [2] to solve the weighted algorithm. The extended algorithm offers the best performance, in terms of statistical accuracy and memory storage, when sufficient a priori information about n is given.
- The new scheme can be used to obtain, in the same generic way, the estimator proposed by [3].

The rest of this paper is organized as follows: In Section 2 we discuss previous works on both the unweighted and weighted cardinality estimation problems. In Section 3 we describe the beta distribution and recall several properties that will be used later. We present our new unified scheme in Sections 4 and 5. In Section 6 we present a weighted version for several known estimators. Finally, in Section 7 we conclude the paper.

2. Related work

State-of-the-art cardinality estimators hash every element e_j into a low dimensional data sketch $h(e_j)$, which can be viewed as a random variable (RV). The different techniques can be classified according to the data sketches they store for future processing. This paper focuses on min/max sketches [2,6,11,15], which store only the minimum/maximum hashed values. The intuition behind such estimators is that each sketch carries information about the desired quantity. For example, when every element e_j is associated with a uniform RV, $h(e_j) \sim \mathrm{U}(0, 1)$, the expected minimum value of $h(e_1), h(e_2), \ldots, h(e_n)$ is $1/(n+1)$. The hash function guarantees that $h(e_j)$ is identical for all the appearances of e_j. Thus, the existence of duplicates does not affect the value of the extreme order statistics. The intuition behind the new unified scheme presented in this paper is that each RV carries information about the weight of the corresponding element, and each sketch carries information about the total weight.

There are other cardinality estimation techniques other than min/max sketches. The first paper on cardinality estimation [7] describes a bit pattern sketch. In this case, the elements are hashed into a bit vector and the sketch holds the logical OR of all hashed values. Bottom-m sketches [4] are a generalization of min sketches, which maintain the m minimal values, where $m \geq 1$. Stable distribution sketches [13] generate a sketch using a vector dot product. A comprehensive overview of the different techniques is given in [2,16].

Previous works have also dealt with the weighted problem. A weighted estimator for continuous variables is given in [3]. Each element is hashed to a random variable derived from exponential distribution $h(e_j) \sim \mathrm{Exp}(w_j)$, where w_j is the weight of e_j. Then, the minimum observed value is stored and used for the estimation. The intuition is that the minimum observed value is exponentially distributed with a parameter that is the weighted sum of the elements. **This estimator can be obtained as a direct result of our unified scheme, when our scheme is applied to continuous max sketches.**

In [5], the weighted problem with integer weights is solved using binary representations. The number of storage units is not fixed, because it depends on the weights. In contrast, the proposed scheme does not assume integer weights, and uses fixed memory. Another weighted estimator, based on continuous bottom-m sketches, is given in [4]. However, bottom-m sketches require maintaining a sorted list of the bottom-m values, which is more computationally demanding than keeping the m separate minimum/maximum values, as in the proposed unified scheme.

3. The beta distribution

We observe that all min/max sketches can be viewed as a two step computation: (a) hash each element uniformly into $(0, 1)$; and (b) store only the minimum/maximum observed value.[1] In the unified scheme we only change step (a) and hash each element into a beta distribution. The parameters of the beta distribution are derived from the weight of the element. In this section, we describe the beta distribution and two of its properties that will be used in the unified scheme.

The Beta(α, β) distribution is defined over the interval $(0, 1)$ and has the following probability and cumulative density functions (PDF and CDF respectively):

$$P\big[X = x \in (0,1)\big] = \frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)\Gamma(\alpha)} x^{\alpha-1}(1-x)^{\beta-1} \tag{1}$$

$$P[X \leq x] = \int_0^x \frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)\Gamma(\alpha)} x^{\alpha-1}(1-x)^{\beta-1} dx, \tag{2}$$

[1] Some estimators (e.g. [6]) transform the uniform hashed values to induce a different distribution, and only then store the minimum/maximum observed value. In Section 6.1 we will consider such estimators.