On Recall and Precision

Conrado Martínez

May 10, 2019

1 Introduction

Consider a (large) data stream $\mathcal{Z} = z_1, \ldots, z_N$, where each z_i is drawn from some domain or *universe* $\mathcal{U}$, and let $n \leq N$ be the number of distinct elements in $\mathcal{Z}$. We may thus look at the multiset X underlying $\mathcal{Z}$

$$X = \{x_1^{f_1}, \ldots, x_n^{f_n}\}$$

where x_1, ..., x_n are the n distinct elements that occur in $\mathcal{Z}$ and f_i denotes the number of occurrences (absolute frequency) of x_i in $\mathcal{Z}$. We will assume, w.l.o.g., that we index the elements in X in non-increasing order of frequency, thus $f_1 \geq f_2 \geq \cdots \geq f_{n-1} \geq f_n > 0$. We will use $p_i = f_i/N$ to denote the relative frequency of x_i.

The two problems that we want to study here are:

1. Top k most frequent elements. Given $\mathcal{Z}$, we want to find $\{x_1, \ldots, x_k\}$ (or any subset of k distinct elements with maximal frequencies).

2. Heavy hitters. Given $\mathcal{Z}$ and a value ϕ, $0 < \phi < 1$, we want to find the $k^* = k^*(\phi)$ distinct elements in $\mathcal{Z}$ with relative frequency $p_i \geq \phi$. Those elements are called *heavy hitters*. Given the data stream and the value ϕ, the largest index k^* such that $p_{k^*} \geq \phi$ is the number of heavy hitters. It is immediate to see that
$$k^* \leq \left\lceil \frac{1}{\phi} \right\rceil$$
If no element has relative frequency above ϕ ($p_1 < \phi$) then we take $k^* = 0$ by convention.

None of these two problems can be solved exactly unless we can keep $\Theta(n)$ elements in memory; thus under the tight memory constraints of the data stream model, we must aim at approximate good solutions.

2 The (ε, δ)-deficient framework

We consider here algorithms (dterministic or randomized) which keep a sample $\mathcal{S}$ of distinct elements from the subsequence of the data stream seen so far.

The algorithms might considerably differ in various aspects, but they will all keep record of apparitions in the data stream of the elements in the sample. In particular, each element kept in the sample will have an associated counter which will be increased each time the corresponding element is seen. That is, if the current instance/apparition z from the data stream is x_i and x_i is currently in the sample the $freq[x_i]$ will be increased by 1. The way that the counters are initialized when an element x is added to the sample, or other updates to these counters (e.g. decrements), will depend on the algorithm. In any case, our algorithms will report several elements in the sample as the answer to a heavy hitters/top k queries and they will also provide an estimate f_i' of the frequency for any of the reported elements x_i, based on the counters $freq[.]$ (f_i' is not necessarily $freq[x_i]$, although this is the case for many algorithms).

2.1 Heavy hitters

- For a deterministic algorithms, we will require that the following conditions are met:

  1. All k^* heavy hitters are reported. That is, if $f_i \geq \phi N$ then x_i is reported as a heavy hitter.
  2. No element x_k such that $p_k < \phi - \varepsilon$ (i.e., $f_k < (\phi - \varepsilon)N$) is reported.
  3. For all reported elements x_i, $f_i - \varepsilon N \leq f_i' \leq f_i$.

- For a randomized algorithm, we have the same conditions happening with high probability (equivalently, the probability that one of the conditions is not met is less than some small δ). Let Y_i denote the indicator random variable for the event "x_i is reported by the algorithm as a heavy hitter", that is, $Y_i = 1$ is x_i is reported by the algorithm and $Y_i = 0$, otherwise. Similarly, let Y_i' be the indicator random variable for the event "x_i is sampled". Notice that we have the stochastic inequality $Y_i \leq Y_i'$: for any instance ω of the probability space in which Y_i and Y_i' are defined we have $Y_i(\omega) \leq Y_i'(\omega)$ since x_i cannot be reported as a relevant element unless x_i is part of the sample. The conditions we require are then:

  1. For all i,
  $$\Pr\{x_i \text{ is reported} \,|\, p_i \geq \phi\} \geq 1 - \delta.$$
  That is, $\mathbb{E}\{Y_i\} \geq 1 - \delta$ for all i, $1 \leq i \leq k^*$.
  2. For all i,
  $$\Pr\{x_i \text{ is reported} \,|\, p_i < \phi - \varepsilon\} < \delta.$$
  Hence, if $p_i < \phi - \varepsilon$ then $\mathbb{E}\{Y_i\} < \delta$.
  3. For all i,
  $$\Pr\{f_i' \in [f_i - \varepsilon, f_i] \,|\, Y_i = 1\} \geq 1 - \delta$$

2.2 Top-k most frequent

We shall use the same definitions as for heavy hitters, taking $\phi := p_k$, that is, the k largest frequency.

3 Precision and recall

In Information Retrieval, *recall* and *precision* are the two most used measures of quality. Recall is the ratio of the number of "relevant" retrieved (=reported) elements to the number of "relevant" elements. Thus, using the indicator variables of the previous section, we will have

$$\mathcal{R} = \frac{Y_1 + \cdots + Y_h}{h}, \tag{1}$$

where $h = k$ (top-k queries) or $h = k^*$ (heavy hitters). Likewise, precision is the ratio of the number of "relevant" retrieved elements to the number of retrieved elements. Hence

$$\mathcal{P} = \frac{Y_1 + \cdots + Y_h}{Y_1 + \ldots + Y_n}. \tag{2}$$

When the algorithms are randomized $\mathcal{R}$ and $\mathcal{P}$ are random variables, where the probability space is that induced by the random choices of the algorithm.

These two mesures have been used in the literature in context of the top-k most frequent and heavy hitters; good algorithms for these problems should exhibit good recall and precison—both close to 1. However, in this context there are two fundamental differences with the typical application in Information Retrieval. In most scenarios of Information Retrieval 1) all elements in the data set are distinct; 2) an element from the data set is either "relevant" or "not relevant".

However, in the problems that we are considering here, the N apparitions in the data stream aren't distinct (at least they are not in the interesting cases!) and not all elements are equally relevant (irrelevant): we should consider an element x_i more "relevant" than another x_j if $f_i \gg f_j$; in other words, if both elements should be reported in the result of a query, an algorithm missing x_j and reporting x_i should score much better than algorithm missing x_i and reporting x_j (and of course both should score worse than an algorithm reporting both!). Moreover, in top-k most frequent queries the algorithm will report k elements from the sample, and thus the number of retrieved elements coincides with the number of relevant documents, i.e., $\mathcal{R} = \mathcal{P}$.

Hence, we will generalize of definition of recall and precision as follows. Pick one element z from the data stream $\mathcal{Z}$ at random, that is, with identical probability $1/N$.

1. Recall:
$$R := \Pr\{z \text{ is retrieved} \mid z \text{ is relevant}\}$$
2. Precision:
$$P := \Pr\{z \text{ is relevant} \mid z \text{ is retrieved}\}$$

Notice that R and P are now conditional probabilities, not random variables.

If h is the largest index of a relevant element ($h = k$ for top-k, $h = k^*$ for heavy hitters) then the numerator is

$$\begin{aligned}\Pr\{z \text{ is relevant and retrieved}\} &= \sum_{i=1}^{n} p_i \Pr\{z \text{ is relevant and retrieved} \mid z = x_i\} \\ &= \sum_{i=1}^{h} p_i \Pr\{x_i \text{ is retrieved}\} = \sum_{i=1}^{h} p_i \mathbb{E}\{Y_i\} = \mathbb{E}\left\{\sum_{i=1}^{h} p_i Y_i\right\},\end{aligned}$$

whereas for the denominators we have

$$\begin{aligned}\Pr\{z \text{ is relevant}\} &= p_1 + \cdots + p_h, \\ \Pr\{z \text{ is retrieved}\} &= \mathbb{E}\left\{\sum_{i=1}^{n} p_i Y_i\right\},\end{aligned}$$

and thus

$$\begin{aligned}R &= \frac{\mathbb{E}\left\{\sum_{i=1}^{h} p_i Y_i\right\}}{p_1 + \cdots + p_h}, \\ P &= \frac{\mathbb{E}\left\{\sum_{i=1}^{h} p_i Y_i\right\}}{\mathbb{E}\left\{\sum_{i=1}^{n} p_i Y_i\right\}}.\end{aligned}$$

Notice that the definition of recall here coincides with the expected value of $\mathcal{R}$ in equation (1) if we set $p_1 = \cdots = p_n = 1/n$ (that is, all elements are considered equally relevant). This is not the case for $\mathcal{P}$; even if we set $p_i = 1/n$ for all i,

$$\frac{\mathbb{E}\left\{\sum_{1\le i\le h} Y_i\right\}}{\mathbb{E}\left\{\sum_{1\le i\le n} Y_i\right\}} \neq \mathbb{E}\{\mathcal{P}\} = \mathbb{E}\left\{\frac{\sum_{1\le i\le h} Y_i}{\sum_{1\le i\le n} Y_i}\right\}.$$

It is also possible to consider slightly different definitions of P and R to take into account that the algorithms answer top-k/heavy hitters queries by selecting a subset of the elements in the sample. The formulas above for P and R do not "penalize" using a lot of extra memory, while they penalize not reporting an element irrespective of whether the element is sampled or not. The alternative formulæ below for R' and P' use the indicators Y_i''s instead of the Y_i's, that is, they correspond to the definition of recall and precision as conditional probabilities replacing "retrieved"(="reported") by "sampled".

$$\begin{aligned}R' &= \frac{\mathbb{E}\left\{\sum_{i=1}^{h} p_i Y_i'\right\}}{p_1 + \cdots + p_h} \\ P' &= \frac{\mathbb{E}\left\{\sum_{i=1}^{h} p_i Y_i'\right\}}{\mathbb{E}\left\{\sum_{i=1}^{n} p_i Y_i'\right\}}\end{aligned}$$

Let us consider now an algorithm that is (ε, δ)-deficient (we may view deterministic algorithms as $(\varepsilon, 0)$-deficient algorithms). A direct consequence of the definition of (ε, δ)-deficiency is that $R \geq 1 - \delta$. It's also true that $\mathbb{E}\{\mathcal{R}\} \geq 1 - \delta$, where $\mathcal{R}$ is the recall as defined by (1). However, the (ε, δ)-deficient framework does not provide useful guarantees for the precision.