Better Space Bounds for Parameterized Range Majority and Minority

Djamal Belazzougui\textsuperscript{1}, Travis Gagie\textsuperscript{1,2}, and Gonzalo Navarro\textsuperscript{3}

\textsuperscript{1} Department of Computer Science, University of Helsinki
\textsuperscript{2} Helsinki Institute for Information Technology
\textsuperscript{3} Department of Computer Science, University of Chile

Abstract. Karpinski and Nekrich (2008) introduced the problem of parameterized range majority, which asks to preprocess a string of length \( n \) such that, given the endpoints of a range, one can quickly find all the distinct elements whose relative frequencies in that range are more than a threshold \( \tau \). Subsequent authors have reduced their time and space bounds such that, when \( \tau \) is given at preprocessing time, we need either \( O(n \lg(1/\tau)) \) space and optimal \( O(1/\tau) \) query time or linear space and \( O((1/\tau) \lg \lg \sigma) \) query time, where \( \sigma \) is the alphabet size. In this paper we give the first linear-space solution with optimal \( O(1/\tau) \) query time. For the case when \( \tau \) is given at query time, we significantly improve previous bounds, achieving either \( O(n \lg \lg \sigma) \) space and optimal \( O(1/\tau) \) query time or compressed space and \( O((1/\tau) \lg (1/\tau) \lg \lg \ln n) \) query time. Along the way, we consider the complementary problem of parameterized range minority that was recently introduced by Chan et al. (2012), who achieved linear space and \( O(1/\tau) \) query time even for variable \( \tau \). We improve their solution to use either nearly optimally compressed space with no slowdown, or optimally compressed space with nearly no slowdown. Some of our intermediate results, such as density-sensitive query time for one-dimensional range counting, may be of independent interest.

1 Introduction

Finding frequent elements in a dataset is a fundamental operation in data mining. Finding the most frequent elements can be challenging when all the distinct elements have nearly equal frequencies and we do not have the resources to compute all their frequencies exactly. In some cases, however, we are interested in the most frequent elements only if they really are frequent. For example, Misra and Gries \cite{MisraGries1973} showed how, given a string and a threshold \( \tau \) with \( 0 < \tau \leq 1 \), with two passes and \( O(1/\tau) \) words of space we can find all the distinct elements in a string whose relative frequencies are at least \( \tau \). These elements are called the \( \tau \)-majorities of the string. Misra and Gries’ algorithm was rediscovered by Demaine, López-Ortiz and Munro \cite{DemaineLopez-OrtizMunro2003}, who noted it can be made to run in \( O(1) \) time per element on a word RAM with \( O(\lg n) \)-bit words, where \( n \) is the length of the string, which is the model we use; it was then rediscovered again by Karp,
Shenker and Papadimitriou [16]. As Cormode and Muthukrishnan [8] put it, “papers on frequent items are a frequent item!”

Krizanc, Morin and Smid [18] introduced the problem of preprocessing the string such that later, given the endpoints of a range, we can quickly return the mode of that range (i.e., the most frequent element). They gave two solutions, one of which takes $O(n^{2-2\epsilon})$ space for any fixed positive $\epsilon \leq 1/2$, and answers queries in $O(n^{\epsilon} \log \log n)$ time; the other takes $O(n^2 \log \log n / \log n)$ space and answers queries in $O(1)$ time. Petersen [22] reduced Krizanc et al.’s first time bound to $O(n^\epsilon)$ for any fixed non-negative $\epsilon < 1/2$, and Petersen and Grabowski [23] reduced the second space bound to $O(n^2 \log \log n / \log^2 n)$. Chan et al. [6] recently gave a linear-space solution that answers queries in $O(\sqrt{n} / \log n)$ time. They also gave evidence suggesting we cannot easily achieve query time substantially smaller than $\sqrt{n}$ using linear space; however, the best known lower bound, by Greve et al. [15], says only that we cannot achieve query time $o((\log(n) / \log(sw/n))$ using $s$ words of $w$ bits each. Because of the difficulty of supporting range mode queries, Bose et al. [5] and Greve et al. [15] considered the problem of approximate range mode, for which we are asked to return an element whose frequency is at least a constant fraction of the mode’s frequency.

Karpinski and Nekrich [17] took a different direction, analogous to Misra and Gries’ approach, when they introduced the problem of preprocessing the string such that later, given the endpoints of a range, we can quickly return the $\tau$-majorities of that range. We refer to this problem as parameterized range majority. Assuming $\tau$ is given when we are preprocessing the string, they showed how we can store the string in $O(n(1/\tau))$ space and answer queries in $O((1/\tau) (\log \log n)^2)$ time. They also gave bounds for dynamic and higher-dimensional versions. Durocher et al. [10] independently posed the same problem and showed how we can store the string in $O(n \log (1/\tau + 1))$ space and answer queries in $O(1/\tau)$ time. Notice that, because there can be up to $1/\tau$ distinct elements to return, this time bound is worst-case optimal. Gagie et al. [14] showed how to store the string in compressed space — i.e., $O(n(H + 1))$ bits, where $H$ is the entropy of the distribution of elements in the string — such that we can answer queries in $O((1/\tau) \log \log n)$ time. They also showed how to drop the assumption that $\tau$ is fixed and simultaneously achieve optimal query time, at the cost of increasing the space bound by a $(\log n)$-factor. That is, they gave a data structure that stores the string in $O(n(H + 1))$ space such that later, given the endpoints of a range and $\tau$, we can return the $\tau$-majorities of that range in $O(1/\tau)$ time. Chan et al. [7] recently gave another solution for variable $\tau$, which also has $O(1/\tau)$ query time but uses $O(n \log n)$ space. As far as we know, these are all the relevant bounds for Karpinski and Nekrich’s original exact, static, one-dimensional problem, both for fixed and variable $\tau$; they are summarized in Table 1, together with our own results. Related work includes Elmasry et al.’s [11] solution for the dynamic version and Lai, Poon and Shi’s [19] and Wei and Yi’s [26] approximate solutions for the dynamic version.

In this paper we first consider the complementary problem of parameterized range minority, which was recently introduced by Chan et al. [7]. For this problem we are asked to preprocess the string such that later, given the endpoints of a