Bottom-k Sketches: Better and More Efficient Estimation of Aggregates

Edith Cohen
AT&T Labs–Research
Florham Park, NJ 07932, USA
edith@research.att.com

Haim Kaplan
School of Computer Science
Tel Aviv University, Tel Aviv, Israel
haimk@cs.tau.ac.il

ABSTRACT

A *Bottom-k sketch* is a summary of a set of items with nonnegative weights. Each such summary allows us to compute approximate aggregates over the set of items. Bottom-k sketches are obtained by associating with each item in a ground set an independent random rank drawn from a probability distribution that depends on the weight of the item. For each subset of interest, the bottom-k sketch is the set of the k minimum ranked items and their ranks. Bottom-k sketches have numerous applications. We develop and analyze data structures and estimators for bottom-k sketches to facilitate their deployment. We develop novel estimators and algorithms that show that they are a superior alternative to other sketching methods in both efficiency of obtaining the sketches and the accuracy of the estimates derived from the sketches.

Categories and Subject Descriptors: G.3;C.2;E.1

General Terms: Algorithms, Measurement, Performance.

Keywords: Sketches, bottom-k, approximate query processing.

1. OVERVIEW

Consider a ground set I of items with weights $w(i) \geq 0$ for $i \in I$. We are interested in obtaining a *sketch* (compact summary or sample) of I that allows us to estimate aggregates over I or over arbitrary subpopulations $J \subset I$ that are specified aposteriori. Sketches of different subsets support estimates of associations and relations (such as size of the intersection) between the subsets. We define *bottom-k* sketches with respect to a family of probability distributions specified by cdf's $\{F_w\}$ ($w \geq 0$). A bottom-k sketch of I is obtained by independently drawing a rank value for each $i \in I$ according to the distribution $F_{w(i)}$ and taking the items with the k smallest rank values. This definition unifies weighted sampling without replacement (using exponentially-distributed rank functions with parameter $w(i)$) [6, 15, 10], unweighted sampling without replacement [3], and priority sampling (using priority ranks) [1].

Bottom-k sketches are an alternative to *k-mins sketches* (see the min-rank [6] and min-hash [4] methods), which consist of the k minimum ranked items in k independent rank assignments. k-mins sketches were used for basic aggregations, including, a random sample and weight estimate, and more complex aggregations such as average weight, approximate quantiles, variance, and higher moments [8]. Sketches of different subsets can be used to estimate their resemblance, weight ratio, and weight of their intersection. In the min-hash [4, 5] method, hash functions replace random rank assignments, the rank assignment of an item depends on the *item identifier*, and it has the property that all copies of the same item obtain the same rank and we can aggregate over *distinct occurrences* without the need for additional book keeping (see [13]). Applications include duplicates detection for Web pages [2, 4], mining of association rules [15] from market basket data, and eliminating redundant network traffic [16].

Bottom-k sketches encode more information than k-mins sketches and for many applications, can be constructed much more efficiently. Our formulation is also more versatile than the k-mins sketches in that it can support arbitrary rank functions. We facilitate the use of bottom-k sketches by providing and analyzing algorithms that construct them and applicable estimators. The benefit of the new estimators is demonstrated by analyzing the variance, constructing confidence intervals, and using simulations on realistic datasets. We provide a brief overview (see [9, 11] for details).

1.1 Efficiency

We propose and analyze data structures and algorithms to construct bottom-k sketches. Existing analysis for k-mins sketches is not applicable, as they are constructed by k independent applications of the same process. We provide tight analysis for both bottom-k and k-mins sketches and show that in most cases the bottom-k alternatives are at least as efficient or considerably more efficient.

Applications maintain sketches of many subsets as items are processed. They manipulate the sketch through two basic operations: A *test* operation which tests if the sketch has to be updated, and an *update* operation which inserts the new item if the sketch indeed has to be updated. We make this distinction since with bottom-k sketches, test operations can be performed much more efficiently than update operations. The number of update operations depends on the order in which items are processed and on the weight distribution of the data. The number of test operations is typically larger than the number of updates, the extent in which it is larger, however, highly depends on the application.

We distinguish between applications with *explicit representation* [4, 2, 15, 16] or *implicit representation* [6, 10, 12] of the data. In applications with an explicit representation, item-subset pairs are provided explicitly. The dataset could be distributed, presented as a data stream, or in external memory, but the pairs are explicitly provided and are all processed to produce the sketches. In applications with implicit representation, the subsets are specified as neighborhoods in a graph or some metric space. With explicit representation, the number of test operations is much larger than the number of update operations.

In addition to plain bottom-k sketches, we consider *all-distances sketches*, that are used when the underlying dataset had items associated with locations in some metric space, and subsets are spec-

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.
ACM 978-1-59593-639-4/07/0006.

ified by neighborhoods of a location. Examples are data streams (where aggregation is over windows of elapsed time to the present time), the Euclidean plane (where we are presented with a query point and distance) [10], a graph (the query is a node and distance) [6, 14], or distributed "spatial aggregation" over a network [6, 10]. An all-distances sketch is a compact encoding of the plain sketches of all neighborhoods of a certain location q. For a given distance d, the sketch for the d-neighborhood of the location can be constructed from the all-distances sketch. We define bottom-k all-distances sketches and present efficient data structures for maintaining both all-distances k-mins sketches and all-distances bottom-k sketches.

For both plain and all-distances sketches, we analyze the number of test and update operations and its dependence on the way the data is represented, the order items are processed, and the distribution of the items' weights. For all-distances sketches, we bound the expected size of the sketches and its dependence on the weight distribution and location of items. For plain sketches, if items have uniform weights or presented in random order, the expected number of updates is $O(k \log n)$, otherwise, there is an additional factor of $\log(\max_i w(i)/\min_i w(i))$ (for certain rank functions). The expected size of all-distances sketches is $O(k \log n)$ if items' weights are uniform or if items are weighted and their location is independent of their weight, and is $O(k \log n \log(\max_i w(i)/\min_i w(i)))$ otherwise. When items are inserted in a sorted order (decreasing rank or distance), the number of updates is equal to the size of the sketch. If items are inserted in a random order, there is an additional factor of $O(\log n)$.

Mimicked sketches k-mins sketches correspond to weighted sampling with replacement whereas bottom-k sketches with exponentially-distributed ranks are weighted samples without replacement. Therefore bottom-k sketches are more informative on distributions where we are likely to obtain repeated samples of the same item (e.g. Zipf). We provide a method to derive k-mins sketches from bottom-k sketches constructed with exponential ranks. Two important benefits of this process are (i) we can use bottom-k sketches in applications (such as those with explicit representation of the data) where they can be obtained much more efficiently than k-mins sketches and still apply existing k-mins estimator and (ii) bottom-k sketch correspond to a distribution over k-mins sketches and therefore the resulting bottom-k estimators are superior to the k-mins estimator in that they have smaller variance. Examples of applications of the mimicking process are provided in the full version.

1.2 Accuracy

Rank-conditioning estimators. For a sketch b, assign to each $i \in b$, an *adjusted weight* $a(i) = w(i)/(1 - F_{w(i)}(r_{k+1}))$, where r_{k+1} is the $(k+1)$th smallest rank value. We prove that for all $i \in I$, $E(a(i)) = w(i)$ (weight for items not in the sketch is 0). Therefore, for any subpopulation $J \subseteq I$, $a(J) = \sum_{i \in J \cap b} a(i)$ is an unbiased estimator of $w(J)$. We also prove that the covariance of the adjusted weight of any two items is zero, and therefore $\mathrm{VAR}(a(J)) = \sum_{i \in J} \mathrm{VAR}(a(i))$. These estimators include priority sampling [1]. The ability to work with arbitrary rank functions allows us to use a single sample for multiple weight functions (such as weight and distinct counts). We also obtain tighter estimators than previously known for weighted sampling without replacement (using $a(i) = w(i)/(1 - \exp(w(i)r_{k+1}))$), which is the only method we are aware of that can be performed efficiently when items are unaggregated [7] (as in IP packet stream or distributed data).

Maximum-likelihood estimators. We use properties of the exponential distribution and careful conditioning to derive the following estimator for weighted sampling without replacement. The estimator is the solution of $\sum_{i=0}^{k} 1/(x - s_i) = r_{k+1}$, where s_i is the sum of the weights of the first i items in b ($s_0 \equiv 0$).

Improved estimators using $w(I)$. Performance of rank conditioning estimators (including priority ranks) suffers in applications where the weight $w(I)$ is easy to obtain and the sketch is used to estimate subpopulations sizes. When $w(I)$ is known, we would like that our estimator to satisfy that $\mathrm{VAR}(a(J)) \ll \sum_{i \in J} \mathrm{VAR}(a(i))$ and in particular $\mathrm{VAR}(a(I)) = 0$. We point on this important issue and derive better estimators that use $w(I)$: (i) The *subset conditioning* estimator for exponential ranks, is unbiased, and superior to the rank-conditioning estimator on *all* subpopulations $J \subset I$, with lower per-item variance and *negative covariances*. With this estimator we have $\mathrm{VAR}(a(I)) = 0$. (ii) The improved maximum-likelihood estimator is the solution of $\sum_{i=0}^{|J \cap b|-1} (x - s_i)^{-1} = \sum_{i=0}^{|(I \setminus J) \cap b|-1} (x - s'_i)^{-1}$, where s_i (resp., s'_i) is the sum of the weights of the first i items in $J \cap b$ (resp., $(I \setminus J) \cap b$.)

Relations and associations. Sketches are used to estimate subset relations such as the intersection, union, or resemblance. Previously, these quantities were computed by applying an estimator to the sketch of the union of the subsets, but this approach effectively utilizes only k out of the (up to) $2k$ samples. We derive rank conditioning and maximum likelihood estimators that utilize all applicable samples. We also derive tighter estimators for subset relations for applications where the total weight of each subset is provided.

Our results are supported by rigorous analysis and their significance is established by simulations. In all, we provide a powerful set of tools for a wide range of applications.

2. REFERENCES

[1] N. Alon, N. Duffield, M. Thorup, and C. Lund. Estimating arbitrary subset sums with few probes. In *Proc. ACM PODS*, pages 317–325, 2005.

[2] K. Bharat and A. Z. Broder. Mirror, mirror on the web: A study of host pairs with replicated content. In *Proc. WWW8*, pages 501–512, 1999.

[3] A. Z. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, pages 21–29. ACM, 1997.

[4] A. Z. Broder. Identifying and filtering near-duplicate documents. In *Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, volume 1848 of *LLNCS*, pages 1–10. Springer, 2000.

[5] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000.

[6] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.*, 55:441–453, 1997.

[7] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup. Sketching unaggregated data streams for subpopulation-size queries. In *Proc. of the ACM PODS*, 2007.

[8] E. Cohen and H. Kaplan. Efficient estimation algorithms for neighborhood variance and other moments. In *Proc. 15th ACM-SIAM Symposium on Discrete Algorithms*. ACM-SIAM, 2004.

[9] E. Cohen and H. Kaplan. Sketches and estimators for subpopulation weight queries. Manuscript, 2007.

[10] E. Cohen and H. Kaplan. Spatially-decaying aggregation over a network: model and algorithms. *J. Comput. System Sci.*, 73:265–288, 2007.

[11] E. Cohen and H. Kaplan. Summarizing data using bottom-k sketches. Manuscript, 2007.

[12] E. Cohen, Y.-M. Wang, and G. Suri. When piecewise determinism is almost true. In *Proc. Pacific Rim International Symposium on Fault-Tolerant Systems*, pages 66–71, December 1995.

[13] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. System Sci.*, 31:182–209, 1985.

[14] H. Kaplan and M. Sharir. Randomized incremental constructions of three-dimensional convex hulls and planar voronoi diagrams, and approximate range counting. In *Proc. ACM SODA*, pages 484–493, 2006.

[15] R. Motwani, E. Cohen, M. Datar, S. Fujiware, A. Gronis, P. Indyk, J. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13:64–78, 2001.

[16] N. T. Spring and D. Wetherall. A protocol-independent technique for eliminating redundant network traffic. In *Proceedings of the ACM SIGCOMM'00 Conference*. ACM, 2000.