Adaptive Algorithms:
Revisiting Klee's measure problem...

Frank Nielsen

11th November 2017

This column is also available in pdf: filename `AdaptiveAlgorithmKleeMeasure.pdf`

In 1977, Klee [4] asked whether the union $\cup_{i=1}^n [a_i, b_i]$ of a set of n intervals can be computed in less than $O(n \log n)$ or not. In general, the *Klee's measure problem* (KMP) asks to calculate the union $\cup_{i=1}^n B_i$ of n axis-parallel boxes (also called isothetic boxes) in $\mathbb{R}^d$. This problem can be solved deterministically in $O(n \log n)$ time for $d \in \{1, 2\}$, and (so far) in $O(n^{\mathrm{floor}}(\frac{d}{2}))$ for arbitrary $d \geq 3$, see Chan [3] (2013).

A simple Monte Carlo algorithm [2] consists in sampling uniformly s points (iid) in the smallest axis-parallel bounding box B of $\cup_{i=1}^n B_i$: The probability of a sample point to fall inside the union $U = \cup_{i=1}^n B_i$ is $\frac{\mathrm{vol}(U)}{\mathrm{vol}(B)}$. Therefore $\mathrm{vol}(U) \simeq \frac{h}{s} \mathrm{vol}(B)$, where h denote the number of points falling in U. This is a probabilistic algorithm that runs in $\tilde{O}(nsd)$ time.

Getting back to Klee's original question: Can we beat the $O(n \log n)$ bound (even in 1D)? This is where two computational aspects pop up: (1) the model of computation, and (2) the concept of adaptive parameter:

1. It is common to consider the *real RAM* (random access machine) model of computation where arithmetic operations are carried in constant time on real numbers (without any precision limitations). If instead, we consider the *word RAM* model [3] (integer input coded using w bits), KMP can be solved in $O\left(\frac{n^{\frac{d}{2}}}{\log^{\frac{d}{2}-2} n} (\log \log n)^{O(1)}\right)$.

2. For special input cases like hypercubes or fat boxes [3], KMP can be solved faster: For example, in $O(n^{\frac{d+1}{3}} \log^{O(1)} n)$ for hypercubes. The 1D (interval) KMP can be solved in $O(n \log p)$ where p denotes the number of piercing points to stab[1] all intervals [8]. Clearly, p is an *adaptive parameter* that depends on the input configuration. So even, if we fix a computation model, there are potentially many adaptive parameters to consider to improve the computational complexity. So a modern extension of Klee's measure problem is to ask whether we can beat the $O(n \log p)$ bound (on real RAM)? Let c denote the number of connected components of $U = \cup_{i=1}^n [a_i, b_i]$. Is it possible to get a $O(n \log c)$ bound. Well, when $p = \frac{n}{2}$ and $c = 1$, we need $O(n \log n)$ time to detect that we have a single component in the union. Indeed, consider the MaxGap problem that consists in finding the largest gap Δ between two consecutive scalars in a given set $\{x_1, \ldots, x_n\}$. Consider the set of intervals

[1]Piercing javascript demo at `https://www.lix.polytechnique.fr/~nielsen/PiercingBoxes/`

$\{B_i = [x_i, x_i + \delta]\}_i$. Then $\cup_{i=1}^n B_i$ has a single component if and only if $\delta \geq \frac{\Delta}{2}$. On the real RAM model of computation, MaxGap has lower bound $\Omega(n \log n)$ (see [9], p. 261). However, by using the floor function and the pigeon principle, one can get a simple linear time algorithm for MaxGap.

It is not easy to find *adaptive (computational) parameters*. For example, consider computing the diameter [5] of a set of n points of $\mathbb{R}^2$. Solving this problem requires $\Omega(n \log n)$-time on the algebraic computation-tree model. However, we can compute the smallest enclosing disk in $\Theta(n)$ time [6]: When a pair of antipodal points are on the border of the smallest enclosing disk, it defines the diameter.

In general, *adaptive algorithms* refine the concept of output-sensitive algorithms by allowing one to take into account further attributes of the input configuration that can be used to improve the overall running time [7] (1996). See also the instance-optimal geometric algorithms [1] (2017).

References

[1] Peyman Afshani, Jérémy Barbay, and Timothy M Chan. Instance-optimal geometric algorithms. *Journal of the ACM (JACM)*, 64(1):3, 2017.

[2] Karl Bringmann and Tobias Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *International Symposium on Algorithms and Computation*, pages 436–447. Springer, 2008.

[3] Timothy M Chan. Klee's measure problem made easy. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 410–419. IEEE, 2013.

[4] Victor Klee. Can the measure of $\cup_{i=1}^n [a_i, b_i]$ be computed in less than $o(nlogn)$ steps? *American Mathematical Monthly*, pages 284–285, 1977.

[5] Grégoire Malandain and Jean-Daniel Boissonnat. Computing the diameter of a point set. *International Journal of Computational Geometry & Applications*, 12(06):489–509, 2002.

[6] Nimrod Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM journal on computing*, 12(4):759–776, 1983.

[7] Frank Nielsen. *Adaptive Computational Geometry ("Algorithmes géométriques adaptatifs")*. Thèse de doctorat en sciences, Université de Nice-Sophia Antipolis, France, 1996.

[8] Frank Nielsen. Fast stabbing of boxes in high dimensions. *Theoretical Computer Science*, 246(1):53–72, 2000.

[9] F.P. Preparata and M.I. Shamos. *Computational geometry: an introduction*. Texts and monographs in computer science. Springer-Verlag, 1988.