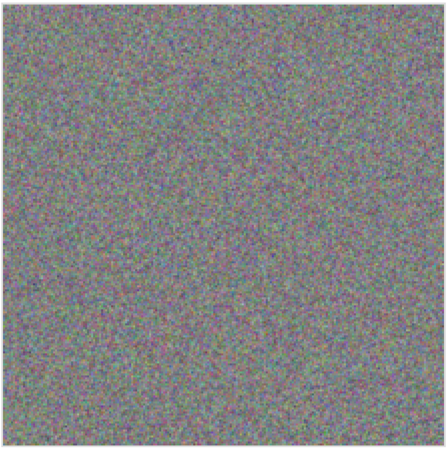
Wikipedia
The Free Encyclopedia

Smoothed analysis

In theoretical computer science, **smoothed analysis** is a way of measuring the complexity of an algorithm. Since its introduction in 2001, smoothed analysis has been used as a basis for considerable research, for problems ranging from mathematical programming, numerical analysis, machine learning, and data mining.[1] It can give a more realistic analysis of the practical performance (e.g., running time, success rate, approximation quality) of the algorithm compared to analysis that uses worst-case or average-case scenarios.



A randomly generated bitmap does not resemble typical pictures.

Smoothed analysis is a hybrid of worst-case and average-case analyses that inherits advantages of both. It measures the expected performance of algorithms under slight random perturbations of worst-case inputs. If the smoothed complexity of an algorithm is low, then it is unlikely that the algorithm will take a long time to solve practical instances whose data are subject to slight noises and imprecisions. Smoothed complexity results are strong probabilistic results, roughly stating that, in every large enough neighbourhood of the space of inputs, most inputs are easily solvable. Thus, a low smoothed complexity means that the hardness of inputs is a "brittle" property.

A typical picture does not resemble a random bitmap.

Although worst-case complexity has been widely successful in explaining the practical performance of many algorithms, this style of analysis gives misleading results for a number of problems. Worst-case complexity measures the time it takes to solve any input, although hard-to-solve inputs might never come up in practice. In such cases, the worst-case running time can be much worse than the observed running time in practice. For example, the worst-case complexity of solving a linear program using the simplex algorithm is exponential,[2] although the observed number of steps in practice is roughly linear.[3][4] The simplex algorithm is in fact much faster than the ellipsoid method in practice, although the latter has polynomial-time worst-case complexity.

Average-case analysis was first introduced to overcome the limitations of worst-case analysis. However, the resulting average-case complexity depends heavily on the probability distribution that is chosen over the input. The actual inputs and distribution of inputs may be different in practice from the assumptions made during the analysis: a random input may be very unlike a typical input. Because of this choice of data model, a theoretical average-case result might say little about practical performance of the algorithm.

Smoothed analysis generalizes both worst-case and average-case analysis and inherits strengths of both. It is intended to be much more general than average-case complexity, while still allowing low complexity bounds to be proven.

History

ACM and the European Association for Theoretical Computer Science awarded the 2008 Gödel Prize to Daniel Spielman and Shanghua Teng for developing smoothed analysis. The name Smoothed Analysis was coined by Alan Edelman.[1] In 2010 Spielman received the Nevanlinna Prize for developing smoothed analysis. Spielman and Teng's JACM paper "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time" was also one of the three winners of the 2009 Fulkerson Prize sponsored jointly by the Mathematical Programming Society (MPS) and the American Mathematical Society (AMS).

Examples

Simplex algorithm for linear programming

The simplex algorithm is a very efficient algorithm in practice, and it is one of the dominant algorithms for linear programming in practice. On practical problems, the number of steps taken by the algorithm is linear in the number of variables and constraints.[3][4] Yet in the theoretical worst case it takes exponentially many steps for most successfully analyzed pivot rules. This was one of the main motivations for developing smoothed analysis.[5]

For the perturbation model, we assume that the input data is perturbed by noise from a Gaussian distribution. For normalization purposes, we assume the unperturbed data $\bar{\mathbf{A}} \in \mathbb{R}^{n\times d}, \bar{\mathbf{b}} \in \mathbb{R}^n, \mathbf{c} \in \mathbb{R}^d$ satisfies $\|(\bar{\mathbf{a}}_i, \bar{b}_i)\|_2 \leq 1$ for all rows $(\bar{\mathbf{a}}_i, \bar{b}_i)$ of the matrix $(\bar{\mathbf{A}}, \bar{\mathbf{b}}).$ The noise $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ has independent entries sampled from a Gaussian distribution with mean 0 and standard deviation σ. We set $\mathbf{A} = \bar{\mathbf{A}} + \hat{\mathbf{A}}, \mathbf{b} = \bar{\mathbf{b}} + \hat{\mathbf{b}}$. The smoothed input data consists of the linear program

maximize

$$\mathbf{c^T} \cdot \mathbf{x}$$

subject to

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}.$$

If the running time of our algorithm on data $\mathbf{A}, \mathbf{b}, \mathbf{c}$ is given by $T(\mathbf{A}, \mathbf{b}, \mathbf{c})$ then the smoothed complexity of the simplex method is[6]

$$C_s(n, d, \sigma) := \max_{\bar{\mathbf{A}}, \bar{\mathbf{b}}, \mathbf{c}} \mathbb{E}_{\hat{\mathbf{A}}, \hat{\mathbf{b}}}[T(\bar{\mathbf{A}} + \hat{\mathbf{A}}, \bar{\mathbf{b}} + \hat{\mathbf{b}}, \mathbf{c})] = \mathrm{poly}(d, \log n, \sigma^{-1}).$$

This bound holds for a specific pivot rule called the shadow vertex rule. The shadow vertex rule is slower than more commonly used pivot rules such as Dantzig's rule or the steepest edge rule[7] but it has properties that make it very well-suited to probabilistic analysis.[8]

Local search for combinatorial optimization

A number of local search algorithms have bad worst-case running times but perform well in practice.[9]

One example is the 2-opt heuristic for the traveling salesman problem. It can take exponentially many iterations until it finds a locally optimal solution, although in practice the running time is subquadratic in the number of vertices.[10] The approximation ratio, which is the ratio between the length of the output of the algorithm and the length of the optimal solution, tends to be good in practice but can also be bad in the theoretical worst case.

One class of problem instances can be given by n points in the box $[0,1]^d$, where their pairwise distances come from a norm. Already in two dimensions, the 2-opt heuristic might take exponentially many iterations until finding a local optimum. In this setting, one can analyze the perturbation model where the vertices $v_1,\dots,v_n$ are independently sampled according to probability distributions with probability density function $f_1,\dots,f_n : [0,1]^d \rightarrow [0,\theta]$. For $\theta = 1$, the points are uniformly distributed. When $\theta > 1$ is big, the adversary has more ability to increase the likelihood of hard problem instances. In this perturbation model, the expected number of iterations of the 2-opt heuristic, as well as the approximation ratios of resulting output, are bounded by polynomial functions of n and θ.[10]

Another local search algorithm for which smoothed analysis was successful is the k-means method. Given n points in $[0,1]^d$, it is NP-hard to find a good partition into clusters with small pairwise distances between points in the same cluster. Lloyd's algorithm is widely used and very fast in practice, although it can take $e^{\Omega(n)}$ iterations in the worst case to find a locally optimal solution. However, assuming that the points have independent Gaussian distributions, each with expectation in $[0,1]^d$ and standard deviation σ, the expected number of iterations of the algorithm is bounded by a polynomial in n, d and σ. [11]

See also

- Average-case complexity
- Pseudo-polynomial time
- Worst-case complexity

References

1. Spielman, Daniel; Teng, Shang-Hua (2009), "Smoothed analysis: an attempt to explain the behavior of algorithms in practice" (http://cs-www.cs.yale.edu/homes/spielman/Research/cacmSmooth.pdf) (PDF), *Communications of the ACM*, **52** (10), ACM: 76–84, doi:10.1145/1562764.1562785 (https://doi.org/10.1145%2F1562764.1562785), S2CID 7904807 (https://api.semanticscholar.org/CorpusID:7904807)

2. Amenta, Nina; Ziegler, Günter (1999), "Deformed products and maximal shadows of polytopes", *Contemporary Mathematics*, **223**, American Mathematical Society: 10–19, CiteSeerX 10.1.1.80.3241 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.80.3241), doi:10.1090/conm/223 (https://doi.org/10.1090%2Fconm%2F223), ISBN 9780821806746, MR 1661377 (https://mathscinet.ams.org/mathscinet-getitem?mr=1661377)
3. Shamir, Ron (1987), "The Efficiency of the Simplex Method: A Survey", *Management Science*, **33** (3): 301–334, doi:10.1287/mnsc.33.3.301 (https://doi.org/10.1287%2Fmnsc.33.3.301)
4. Andrei, Neculai (2004), "Andrei, Neculai. "On the complexity of MINOS package for linear programming", *Studies in Informatics and Control*, **13** (1): 35–46
5. Spielman, Daniel; Teng, Shang-Hua (2001), "Smoothed analysis of algorithms", *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, ACM, pp. 296–305, arXiv:cs/0111050 (https://arxiv.org/abs/cs/0111050), Bibcode:2001cs.......11050S (https://ui.adsabs.harvard.edu/abs/2001cs.......11050S), doi:10.1145/380752.380813 (https://doi.org/10.1145%2F380752.380813), ISBN 978-1-58113-349-3, S2CID 1471 (https://api.semanticscholar.org/CorpusID:1471)
6. Dadush, Daniel; Huiberts, Sophie (2018), "A friendly smoothed analysis of the simplex method", *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 390–403, arXiv:1711.05667 (https://arxiv.org/abs/1711.05667), doi:10.1145/3188745.3188826 (https://doi.org/10.1145%2F3188745.3188826), ISBN 9781450355599, S2CID 11868079 (https://api.semanticscholar.org/CorpusID:11868079)
7. Borgwardt, Karl-Heinz; Damm, Renate; Donig, Rudolf; Joas, Gabriele (1993), "Empirical studies on the average efficiency of simplex variants under rotation symmetry", *ORSA Journal on Computing*, **5** (3), Operations Research Society of America: 249–260, doi:10.1287/ijoc.5.3.249 (https://doi.org/10.1287%2Fijoc.5.3.249)
8. Borgwardt, Karl-Heinz (1987), *The Simplex Method: A Probabilistic Analysis* (https://nbn-resolving.org/urn:nbn:de:bvb:384-opus4-143220), Algorithms and Combinatorics, vol. 1, Springer-Verlag, doi:10.1007/978-3-642-61578-8 (https://doi.org/10.1007%2F978-3-642-61578-8), ISBN 978-3-540-17096-9
9. Manthey, Bodo (2021), Roughgarden, Tim (ed.), "Smoothed Analysis of Local Search" (https://www.cambridge.org/core/books/beyond-the-worstcase-analysis-of-algorithms/smoothed-analysis-of-local-search/CA67DD5FE32ABD53898165847C3F86C7), *Beyond the Worst-Case Analysis of Algorithms*, Cambridge: Cambridge University Press, pp. 285–308, doi:10.1017/9781108637435.018 (https://doi.org/10.1017%2F9781108637435.018), ISBN 978-1-108-49431-1, S2CID 221680879 (https://api.semanticscholar.org/CorpusID:221680879), retrieved 2022-06-15
10. Englert, Matthias; Röglin, Heiko; Vöcking, Berthold (2007), "Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP", *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, **68**: 190–264, arXiv:2302.06889 (https://arxiv.org/abs/2302.06889), doi:10.1007/s00453-013-9801-4 (https://doi.org/10.1007%2Fs00453-013-9801-4)
11. Arthur, David; Manthey, Bodo; Röglin, Heiko (2011), "Smoothed Analysis of the k-Means Method" (https://ris.utwente.nl/ws/files/6539396/JACM_ArthurEA_kMeansSmoothed.pdf) (PDF), *Journal of the ACM*, **58** (5): 1–31, doi:10.1145/2027216.2027217 (https://doi.org/10.1145%2F2027216.2027217), S2CID 5253105 (https://api.semanticscholar.org/CorpusID:5253105)