Smoothed Analysis in Learning: Tensors and k-Means

Hunter Lang
MIT
hjl@mit.edu

Carlos Cortez
MIT
cortezc@mit.edu

Abstract

Most learning problems are hard in the worst-case, so much of the current research focuses on finding good heuristics, polynomial-time approximation algorithms, or special cases with proveable accuracy and runtime guarantees. But many algorithms that are inefficient in the worst case (Simplex being the classical example) consistently seem to run fast in practice. Smoothed analysis gives a framework for better understanding performance on real-world data, specifically for problems where some component is not adversarial. This is a natural assumption in learning, since data in the learning setting are usually prone to measurement or modeling noise. We survey the (very much ongoing) application of smoothed analysis to learning problems by way of two examples: k-means and tensor decomposition.

1. Introduction

Spielman and Teng [1], [2] introduced smooth analysis to give a more suitable framework for predicting real-life algorithm performance. Not every algorithm that runs fast in practice is polynomial-time; worst-case analysis falls short of explaining why some "slow" algorithms are actually quite efficient in practice. The original paper [1] gave a proof that the Simplex algorithm has polynomial "smoothed complexity": that is, if you assume the data are subject to random perturbations (the kind that would arise from measurement noise in practice), Simplex runs in expected polynomial time. This sparked a host of smoothed analysis proofs involving classic combinatorial optimization problems. [EXAMPLES]. The key assumption inherent in applying smoothed analysis techniques to an algorithm is that some component of the problem is not adversarial. The hope is that worst-case inputs are somehow isolated in the input space (indeed, the worst-case inputs to many algorithms are intricate and fragile), so that any real data will, with high probability, not be worst-case. In recent years, there has been an increasing trend of applying smoothed analysis to learning problems [2], [4], [**?**], [**?**].

2. Tensor Decomposition

2.1. Algorithm

2.2. Worst-case

2.3. Smoothed Analysis

3. Conclusion

References

[1] Daniel A. Spielman and Shang-Hua Teng, "Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually takes Polynomial Time." *Journal of the ACM, Vol 51 (3)*, pp. 385 - 463, 2004.

[2] Daniel A. Spielman and Shang-Hua Teng, "Smoothed Analysis: An Attempt to Explain the Behavior of Algorithms in Practice" *Communications of the ACM, Vol. 52 No. 10*, pp. 76-84, 2009.

[3] Andrea Vattani, "k-means Requires Exponentially Many Iterations Even in the Plane." *Proc. of the 25th ACM Symp. on Computational Geometry (SoCG)*, pp 324332, 2009.

[4] David Arthur, Bodo Manthey, and Heiko Roeglin, "Smoothed Analysis of the k-means Method." 2010.

[5]

[6] Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan, "Smoothed Analysis of Tensor Decomposition", CoRR, 2013.