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 provable accuracy and runtime guarantees. But many algorithms that are inefficient in the worst case consistently seem to run fast in practice (Simplex being the classical example). 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 smoothed 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 empirically quite efficient. As a first application of their techniques, 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 noice (the kind that would arise in many practical scenarios), Simplex runs in expected polynomial time. This sparked a host of papers applying smoothed analysis to classic combinatorial optimization problems. [EXAMPLES]. The key assumption of the smoothed analysis setting is that some component of the problem is not adversarial. The hope is that worst-case instances are somehow isolated in the input space (indeed, the worst-case inputs to many algorithms are intricate and fragile), so that any real data is unlikely to be a worst-case instance. In recent years, there has been an increasing trend of applying smoothed analysis to learning problems [2], [4], [5], [6]. Some researchers have followed the standard line of smoothed analysis, focusing on algorithm *runtimes*. Others have expanded the ideas of smoothed analysis to show that algorithms still work under noise. We will explain the difference by an example.

2. Tensor Decomposition

Among other problems, tensor decomposition methods can be used to learn topic models, multi-view mixture models, phylogenetic trees, and detect communities [CITE–course monograph?]. In practice, tensors we want to decompose into factors are random.

3. k-means

k-means is an algorithm

4. 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] Adam Tauman Kalai, Alex Samorodnitsky, and Shang-Hua Teng, "Learning and Smoothed Analysis", *IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 395-404, 2009.

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