Differentially private algorithm configuration[*]

Maria-Florina Balcan Travis Dick Ellen Vitercik

August 16, 2017

Introduction. Algorithms regularly depend on parameters effecting run-time and solution quality. Researchers have developed machine learning techniques for automated parameter tuning which use a training set of problem instances to determine a configuration with high expected performance over future instances. This line of work has inspired breakthroughs in diverse fields including combinatorial auctions [10], scientific computing [4], vehicle routing [3], and SAT [14]. Since the resulting configuration depends on the training set, it might leak sensitive information about problem instances contained therein. We provide a general framework for differentially private automated algorithm configuration.

We apply our algorithmic framework to many problems where privacy preservation is essential, such as integer quadratic programming (IQP). We study algorithm configuration for a well-known family of IQP approximation algorithms based on semidefinite programming [6, 16]. IQPs appear frequently in machine learning applications, such as MAP inference [8, 15, 5], community detection [2], variational methods for graphical models [12], and graph-based semi-supervised learning [13]. We also apply our algorithm to many greedy algorithm families, including algorithms for the knapsack problem and maximum weight independent set problem.

Our main algorithm. We model an application domain as a distribution $\mathcal{D}$ over problem instances [1, 7]. A parameter's performance is the expected value of a user-defined *utility function* $U(x, \rho)$ mapping a problem instance x and parameter ρ to a real-valued score. Our algorithm returns a parameter $\hat{\rho}$ that approximately maximizes $U(S, \rho)$, the average utility of a parameter ρ over a sample S, while preserving differential privacy. Under reasonable assumptions, as the sample size grows, the expected utility of $\hat{\rho}$ approaches the optimal parameter's expected utility.

Our algorithm is inspired by a phenomenon commonly observed in the algorithm configuration literature (e.g., [1, 7]): given a sample S, $U(S, \rho)$ is a piecewise constant function of ρ with discontinuities at a few *critical points*. Our algorithm returns each critical point ρ with probability proportional to $\exp(\epsilon|S|U(S, \rho)/H)$, as in the classic *exponential mechanism* [11]. It succeeds when these critical points are well-dispersed because as we range over ρ, the parameterized algorithm's behavior changes on exactly one problem instance in the training set when ρ crosses a critical point. Therefore, given two values ρ and ρ', the difference in average utility between them scales linearly with the number of critical points between them. Intuitively, if the critical points concentrate near the optimal parameter value ρ^*, then even if only a small amount of noise is added to the algorithm's output $\hat{\rho}$, there may be many critical points separating $\hat{\rho}$ and ρ^*, possibly leading to severe suboptimality.

[*] Authors' addresses: {ninamf, tdick, vitercik}@cs.cmu.edu.

We prove that if at most k critical points fall in any interval of width w, the utility of the configuration our algorithm returns is at most $\frac{H}{|S|}\left(\frac{2\log(\lceil 1/w\rceil)}{\epsilon}+k\right)$ from the optimal configuration's utility, where H upper bounds the range of U over the support of $\mathcal{D}$.

Applications. When we apply our algorithm to a family of IQP approximation algorithms [6, 16], the utility function corresponds to the objective value of the solution returned by the algorithm parameterized by ρ. We prove that our configuration algorithm determines a parameter with utility that is within $\tilde{O}\left(\frac{H}{|S|}\left(\frac{1}{\epsilon}+n\sqrt{|S|}\right)\right)$ of the utility of the optimal parameter while preserving ϵ-differential privacy. We also apply our main algorithm to several parameterized classes of greedy algorithms and arrive at similarly strong utility guarantees.

Related work. Kusner et al. present a differentially private Baysian optimization algorithm with applications to hyper-parameter tuning [9]. It succeeds under various assumptions which are significantly different from ours, for example that a configuration's performance is Lipschitz in its parameters, which does not hold for the problems we study.

Acknowledgments

This work was supported in part by NSF grants CCF-1422910, CCF-1535967, IIS-1618714, and a Microsoft Research Faculty Fellowship.

References

[1] Maria-Florina Balcan, Vaishnavh Nagarajan, Ellen Vitercik, and Colin White. Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems. *Proceedings of the Conference on Learning Theory (COLT)*, 2017.

[2] Afonso S. Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 361–382, 2016.

[3] Yves Caseau, François Laburthe, and Glenn Silverstein. A meta-heuristic factory for vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming (CP)*, pages 144–158. Springer, 1999.

[4] Jim Demmel, Jack Dongarra, Victor Eijkhout, Erika Fuentes, Antoine Petitet, Rich Vuduc, R Clint Whaley, and Katherine Yelick. Self-adapting linear algebra algorithms and software. *Proceedings of the IEEE*, 93(2):293–312, 2005.

[5] Roy Frostig, Sida Wang, Percy S Liang, and Christopher D Manning. Simple MAP inference via low-rank relaxations. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3077–3085, 2014.

[6] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[7] Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection. In *Proceedings of the ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 123–134. ACM, 2016.

[8] Qixing Huang, Yuxin Chen, and Leonidas Guibas. Scalable semidefinite relaxation for maximum a posterior estimation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 64–72, 2014.

[9] Matt Kusner, Jacob Gardner, Roman Garnett, and Kilian Weinberger. Differentially private Bayesian optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 918–927, 2015.

[10] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Empirical hardness models: Methodology and a case study on combinatorial auctions. *Journal of the ACM (JACM)*, 56(4):22, 2009.

[11] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007.

[12] Andrej Risteski and Yuanzhi Li. Approximate maximum entropy principles via Goemans-Williamson with applications to provable variational methods. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 4628–4636, 2016.

[13] Jun Wang, Tony Jebara, and Shih-Fu Chang. Semi-supervised learning using greedy max-cut. *Journal of Machine Learning Research*, 14(Mar):771–800, 2013.

[14] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32:565–606, June 2008.

[15] Mingjun Zhong, Nigel Goddard, and Charles Sutton. Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3590–3598, 2014.

[16] Uri Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 1999.