Ioannis Mitliagkas — Research Statement

Modern data-driven applications rely on skilled data scientists—people with full command of known statistical guarantees and efficient computation techniques—and powerful computational infrastructure to deliver analytical insights. Big organizations like web companies, retailers and government have the resources to attract experts and maintain powerful infrastructure. Social groups, NGOs and academic research units, on the other hand, lack funding and struggle to attract the people and maintain infrastructure.[1] Brokers like DataKind connect data scientists to organizations of impact, but depend on data scientists volunteering their time while being aggressively recruited in industry. To make data analysis more accessible, it is necessary that we bridge the gap between tools, human capital and infrastructure. This is an ambitious vision, but we are already taking steps towards it.

My past and current research, as well as the directions I wish to pursue in the future, touch upon themes of *optimization, analytical guarantees and tuning*, all necessary for bringing this vision about. Work towards this goal can be broken into three major thrusts.

1. **Optimize** popular tools to make them resource-efficient. This includes work on faster algorithms and systems that perform well with less computation and less data, lowering the cost of entry for small groups. A big part of my research effort goes into understanding the fundamental limits of popular techniques and coming up with resource-light alternative algorithms and systems.
2. **Automate** the tuning and assembly of data analytics pipelines to make them usable by non-experts. Tuning commonly used methods, and complex systems like in deep learning, requires work on understanding their *computational and statistical behavior* and providing *data-dependent guarantees*—a major focus of my research. The automated assembly of large models and analytics pipelines further requires a deep understanding of the *dynamics and interactions in complex systems*; this is a direction I am interested in exploring further.
3. **Educate** the new generation of citizens that will be capable of using this friendlier generation of tools. My major teaching aspiration is to help prepare non-Ph.D.-level data scientists.

Research Overview

My research is focused on theory, algorithms and systems in the intersection of statistics and computation. It spans high-dimensional statistics, machine learning, optimization, statistical inference and large-scale distributed systems. I strive to extend our current understanding of ubiquitous tools and optimize their use either through new theoretical guarantees or seemingly minor algorithmic and systems tweaks that can deliver unexpected benefits. Interaction with research labs in industry and academia drives my research by closing the loop with applications and offers opportunities for large-scale deployment of my recent work. Main themes of my work:

New insights on simple, classic tools. There is good reason why classic tools are established. Methods like principal component analysis [MCJ13], PageRank [MBDC14], stochastic gradient descent (SGD) [MZHR16, ZDSMR16] and Gibbs sampling [HDSMR16] are ubiquitous in their respective tasks, because they have proved their merit time and again. Furthermore, inertia prevents the adoption and deployment of new methods and systems, especially if they are unnecessarily complex and do not deliver a serious breakthrough in performance. In my work I look at classic and massively deployed tools from a new perspective, providing new guarantees or minor tweaks that extend their use cases, improve their performance or reduce their resource footprint. The end result is research with a high potential for impact.

Systems. Theoretical and algorithmic breakthroughs have a bigger impact when translated into a real system. I find that working on a system prototype is useful as a proof of concept, as a source of new technical challenges and research questions, and as an experimentation platform. In [PMDC14], we implement and deploy our graph algorithm on hundreds of AWS nodes using MapReduce. In [MBDC14], we hack GraphLab to improve its PageRank performance. In Omnivore [HZMR16], we put our recent theory to use [MZHR16] and come up with a prototype deep learning system that is an order of magnitude faster than state-of-the-art systems.

[1] https://hbr.org/2014/08/recruiting-data-scientists-to-do-social-good

Beyond worst-case guarantees. My research usually forgoes *worst-case analysis*. Seen as the ultimate guarantee, worst-case bounds are sought after by the theory community and often give great insights into a problem. They are, however, usually pessimistic. Systems behave in some *typical* manner. Furthermore, *data-dependent guarantees* are usually much stronger. In [PMDC14] we give graph-dependent guarantees for recovering dense k-subgraphs, and in [MCJ13] we give the first guarantees for streaming PCA in a noisy setting.

Relationship with industry and research labs. Interaction with labs involved in applied research and product development is a valuable source of technical challenges and, sometimes, funding. My work on asynchronous deep learning resulted into a joint project with Intel and NERSC on a planned submission to Supercomputing'17, aiming for the Gordon Bell Prize. It also led to visits and discussions with nVidia, Google, MIT Lincoln Labs and Los Alamos National Labs, which gave rise to new, interesting theoretical questions, like the analysis of classical methods using the theory of orthogonal polynomials. My deep learning work also led to a collaboration with Dr. Daniel Rubin's Stanford Radiology lab, that focuses on applying our technology to histopathology and bone-tumor identification tasks. During my Ph.D. studies, I had the opportunity to interact with Teradata and booking.com, which motivated me to work on hacking GraphLab to improve its PageRank performance.

Past and Current Work Highlights

Summary. During my postdoctoral work, I discovered a new connection between asynchrony and momentum with direct implications for the performance and tuning of existing systems deployed by Google, Intel, nVidia and others. Our prototype deep learning system uses this theoretical understanding to outperform the state-of-the-art by almost an order of magnitude. I coauthored a paper dispelling a commonly held conjecture regarding the relative performance of different scan orders in Gibbs sampling and another one providing understanding on when frequent model averaging benefits parallel SGD. During my Ph.D., I came up with a novel analysis and guarantees for Streaming PCA, designed new algorithms for finding dense subgraphs and deployed them on a cluster of 100 machines using MapReduce, extended the notion of typicality to give strong converse results for inverse problems, analyzed dependent random walks and hacked the GraphLab codebase to support randomized algorithms improving its PageRank performance by an order of magnitude.

Theory

Asynchrony Induces Momentum [MZHR16]. We show that running SGD in an asynchronous manner can be viewed as adding a momentum-like term to the SGD iteration. Our result does not assume convexity of the objective function, so it is applicable to deep learning systems. An important implication is that tuning the momentum parameter is important when considering different levels of asynchrony, and that recent results by big groups like Google's TensorFlow missed this key optimization and report suboptimal results for asynchronous configurations. Finally, our theory suggests new ways of counteracting the adverse effects of asynchrony: for example, using *negative algorithmic momentum* can improve performance under high asynchrony. Our results have gained attention from Google, Microsoft, nVidia and Intel, as well as a number of national labs.

Scan order in Gibbs sampling is important [HDSMR16]. There are two common scan orders: random scan and systematic scan. It has been conjectured that the mixing times of random scan and systematic scan do not differ by more than a logarithmic factor. We show by counterexample that this is not the case, and prove that that the mixing times do not differ by more than a polynomial factor under mild conditions. To prove these relative bounds, we introduce a method of augmenting the state space to study systematic scan using conductance.

Model averaging in parallel SGD [ZDSMR16]. Periodic model averaging is a common but not well understood practice. We study model averaging as a variance-reducing mechanism and describe two ways in which the frequency of averaging affects convergence. For convex objectives, we show the benefit of frequent averaging depends on the gradient variance envelope. For non-convex objectives, we illustrate that this benefit depends on the presence of multiple globally optimal points.

Streaming PCA [MCJ13]. From known phase transition results we know that in the *noisy setting* the number of samples required to recover principal components is $O(p)$. This means that a batch algorithm requires $O(p^2)$ memory and storage and motivates a *memory-limited, single-pass streaming* algorithm. Our work in [MCJ13] was the first to provide an algorithm along with global convergence guarantees and tight characterization of the sample complexity for the streaming PCA problem. It is easily parallelizable, and our multicore implementation with nearly linear scalability can be found in [Mit14]. It can handle an overwhelming number of sample entry erasures [MCJ15]. Our algorithm has been implemented at least four times by Julia and R developers[2].

Information Theoretic Bounds and Learning Rankings [MV10, MGCV11]. The machine learning community uses tools like oracle analysis and Cramér-Rao bounds to reason about fundamental (or "minimax") limits. Information theory has some invaluable tools to offer, that are often overlooked. Fano's inequality yields what is called a *weak converse theorem*. That means, if problem-specific necessary conditions are not met, then the probability of recovery error is bounded away from zero, for any recovery method. Our work [MV10] introduces different methodology for providing *strong* converse theorems for general inverse problems. The strength of the results lies in the fact that the probability of error is actually shown to converge to one—a guaranteed disaster—if the necessary conditions are not met. In [MGCV11] we use similar tools to provide tight achievability and strong converse results for the task of learning a number of rankings, jointly from many users' pairwise preferences.

System Application

Omnivore: tuning deep learning systems [HZMR16]. In this systems paper, we study the factors affecting training time in multi-device deep learning systems. We show that in the single-node setting throughput can be improved by at least 5.5× over state-of-the-art systems on CPUs. We use our novel understanding of the interaction between system and optimization dynamics to provide an efficient hyperparameter optimizer. Our optimizer involves a predictive model for the total time to convergence and selects an allocation of resources to minimize that time. By doing this optimization, our prototype runs 1.9× to 12× faster than the fastest state-of-the-art systems. Following up on this work, we have an ongoing collaboration with Intel, to implement our features of asynchronous-aware tuning and hybrid asynchronous configurations into IntelCaffe (already in production), and with NERSC for its planned submission for the Gordon Bell prize.

Fast PageRank Approximations on Graph Engines [MBDC14]. Motivated by discussions with people in the industry, we worked on improving PageRank performance on popular *graph engines*,and focus on GraphLab as it has proven to be the fastest among distributed engines. Our algorithm uses random walks as a natural discretization of the power method. The biggest component of our work is a modification of the GraphLab engine to allow for *randomized synchronization* between the graph nodes [MBDC14]. This change reduces the communication requirements of the algorithm significantly. As a side-effect, random walks on the graph are not independent anymore, posing an analytical challenge. We nonetheless demonstrated experimentally and analytically that our method gives a 7x to 10x of speed improvement over the state of the art (vanilla GraphLab).

Densest k-Subgraphs on MapReduce [PMDC14]. Given a graph and integer parameter k, our objective is to find a subgraph consisting of k nodes and containing as many edges as possible. The problem is NP-hard and hard to approximate. These results are discouraging, but they pertain to the *worst case*. A central theme of my work is skipping worst-case analysis in favour of typical and data-dependent analysis. In [PMDC14] we do that. Using a low-rank approximation for the adjacency matrix, and leveraging combinatorial methods for quadratic optimization over low-rank spaces we get an efficient solver for this problem. Furthermore, the spectral decomposition, provides us with a *data-dependent* bound on the quality of the solution. This bound comes as a free by-product of the solver and, in practice, it told us that in all of our experiments we achieved at least 70% of the optimum density. Using our MapReduce implementation we solved for graphs with billions of edges.

[2] https://libraries.io/github/eric-tramel/StreamingPCA.jl, https://rdrr.io/cran/onlinePCA/man/bsoipca.html, https://cran.r-project.org/web/packages/onlinePCA/onlinePCA.pdf, http://finzi.psych.upenn.edu/library/onlinePCA/html/bsoipca.html

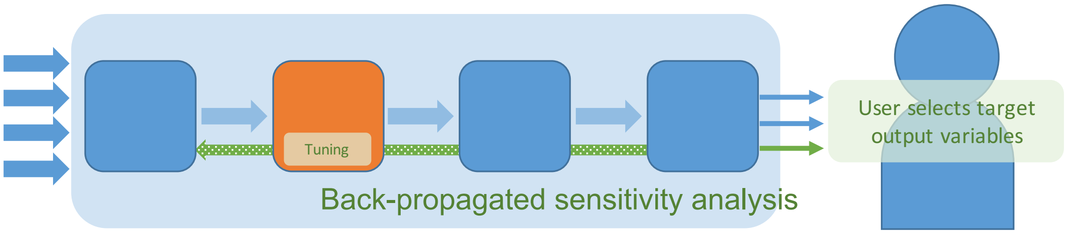
Tuning
User selects target output variables
Back-propagated sensitivity analysis

Figure 1: Motivating vision: Data analyst provides data sources, is guided through the assembly of a data analysis pipeline and sets constraints on the quality of the output. Tradeoff-aware system "back-propagates" constraints throughout the pipeline, tuning individual components to optimize performance.

Future Work

I plan to continue a mixture of theoretic, algorithmic and systems work to make data analysis resource-efficient and simple to use. Part of the automation vision is described in Figure 1; my goal is to keep producing the various elements that can make it a reality. New guarantees and optimizations for learning and inference tools as well as understanding systems tradeoffs are necessary for automated assembly and tuning of data pipelines.

Exact analysis for optimization Much of the existing analysis and guarantees use a series of bounds that yield good-enough, or even tight results, but often obscure intuition. Nesterov's accelerated gradient method is a classic example of analysis that "just works," but has otherwise resisted attempts to extracting intuition. As seen in my previous work, there are often overlooked insights hiding in popular methods, waiting to be discovered. Intuition can come from theory and reveal unexpected phenomena and prescribe strategies, like using negative momentum to compensate for the effects of asynchrony [MZHR16]. Using exact tools, like families of orthogonal polynomials, to analyze system and optimization dynamics is a promising direction with great expository value. Polynomial families can be used to capture the dynamics and analyze tools like Krylov methods for PCA, accelerated optimization and spectrum estimation and, sometimes, reveal previously unknown phenomena.

Targeted, optimized inference Gibbs sampling successively simulates variables from the univariate conditionals of a multivariate target distribution. The principal degree of freedom there is the *scan*, the order in which each variable is sampled. In [HDSMR16] we studied the relative efficacy of *systematic*, and *uniform random scans*. Non-uniform scans, however, can lead to more accurate inferences both in theory and in practice. This effect is particularly pronounced when certain variables are of greater inferential interest. Pipelines that include inference components can benefit by the development of efficient procedures for optimizing the scan order.

Automated tuning and assisted pipeline assembly Beyond improvements in individual learning, optimization and inference components, I also want to study their interactions when used in common pipeline configurations. Understanding how these dynamics interact can drastically reduce the hyperparameter space [HZMR16] and help us tune the pipeline as a whole. Pipeline-wide data-dependent guarantees pose an interesting challenge. Simple ideas like back-propagating quality constraints—imposed by the user on the output—through the pipeline can create coupled constraints and provide the opportunity of pipeline-wide tuning. Taking things a step further, we can envision a system that helps non-experts with assisted model assembly. Starting with prescriptions for standard use cases, and using recent results and guarantees from adaptive data analysis, we can help the user iterate over a different pipelines until good results are achieved, while controlling overfitting.

Applications Work on applications puts theoretical and systems advances into use, while informing next research steps. My collaboration with Lawrence Livermore National Labs, involves applying our deep learning technology on high energy physics and climate applications. In my ongoing collaboration with Dr. Daniel Rubin's Radiology Lab at Stanford, we apply efficient deep learning, semi-supervised learning and domain adaptation techniques to data-limited tasks from histopathology and oncology, like identifying bone tumours from X-rays given only only a small labeled sample and a large unlabeled data set.

References

[HDSMR16] Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Scan order in gibbs sampling: Models in which it matters and bounds on how much. *NIPS*, 2016.

[HZMR16] Stefan Hadjis, Ce Zhang, Ioannis Mitliagkas, and Christopher Ré. Omnivore: An optimizer for multi-device deep learning on cpus and gpus. *Under Review*, 2016.

[MBDC14] Ioannis Mitliagkas, Michael Borokhovich, Alex Dimakis, and Constantine Caramanis. FrogWild! fast pagerank approximations on graph engines. *VLDB*, 2014.

[MCJ13] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory Limited, Streaming PCA. In *Advances in Neural Information Processing Systems*, pages 2886–2894, 2013.

[MCJ15] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Streaming PCA with Many Missing Entries. *Preprint*.

[MGCV11] Ioannis Mitliagkas, Aditya Gopalan, Constantine Caramanis, and Sriram Vishwanath. User rankings from comparisons: Learning permutations in high dimensions. In *Proc. of Allerton Conf. on Communication, Control and Computing, Monticello, USA*, 2011.

[Mit14] Ioannis Mitliagkas. Pyliakmon streaming library. `https://github.com/migish/pyliakmon`, 2014. Accessed: 2014-03-26.

[MV10] Ioannis Mitliagkas and Sriram Vishwanath. Strong information-theoretic limits for source/model recovery. In *Proc. of Allerton Conf. on Communication, Control and Computing, Monticello, USA*, 2010.

[MZHR16] Ioannis Mitliagkas, Ce Zhang, Stefan Hadjis, and Christopher Ré. Asynchrony begets momentum, with an application to deep learning. *Appeared in Proc. of Allerton Conf. on Communication, Control and Computing, Monticello, USA*, 2016.

[PMDC14] Dimitris Papailiopoulos, Ioannis Mitliagkas, Alexandros Dimakis, and Constantine Caramanis. Finding dense subgraphs via low-rank bilinear optimization. In *ICML 2014*, pages 1890–1898, 2014.

[ZDSMR16] Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Parallel SGD: When does averaging help? *OptML: Optimization Methods for the Next Generation of Machine Learning, workshop at ICML*, 2016.