Ioannis Mitliagkas
Research Statement

Large-scale learning problems pose exciting algorithmic, analytic and computational challenges. Often, there exist established – sometimes exact – algorithms. Their *resource requirements*, however, limit their scalability. Exhaustive combinatorial algorithms hit computational bottlenecks even for small/medium problem sizes. More efficient algorithms avoid this issue, only to soon hit a memory bottleneck as the parameter space grows too big. This is true for algorithms depending on the estimation of second (or higher) order statistics and for algorithms that perform join operations on moderately sized datasets. Finally, algorithms avoiding this pitfall are met with local storage limitations. This means that the machine responsible for the computation does not have the capacity to store the full dataset.

In my research I focus on these issues – as they arise from learning problems – and find provably efficient solutions. The ideal research project involves a mixture of clever algorithmic design, an actual system implementation and tight supporting analysis.

With focus on application, my research usually forgoes dealing with *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. Most real systems behave in some *typical* manner. For non-critical applications the rare departure from that typical behaviour is not catastrophic. Furthermore, *data-dependent guarantees* are usually much stronger.

Whenever possible, we use tools "off the shelf." It is common however, that we need to develop new tools for the problem at hand. For example, in my work, I came up with a novel analysis for Streaming PCA, designed new algorithms for finding dense subgraphs and deployed them on a cluster of 100 Amazon EC2 machines, extended the notion of typicality to give strong converse results for inverse problems, analyzed dependent random walks and rewrote the GraphLab codebase and API to support randomized algorithms. The following sections discuss some of the problems that interest me and give examples of the main themes introduced so far. Each section concludes with a short outline on promising directions and themes I intend to explore in the next few years.

Streaming PCA

Principal Component Analysis (PCA) [Jolo5] is the problem of finding the *dominant directions* present in a set of p-dimensional vectors. The problem is posed either in terms of variance maximization, or as subspace recovery [XCS10]. The most successful batch algorithm eigen-decomposes the sample covariance matrix to get the principal directions [Jolo5]. This algorithm is statistically optimal [BBAP05], but requires $O(p^2)$ memory to store the covariance matrix estimate. Hence, it hits the memory bottleneck discussed in the introduction. Furthermore, from the results in [BBAP05] we gather that in the *noisy setting* the number of samples required to recover the principal components is $O(p)$. This means that a batch algorithm requires storage $O(p^2)$ and, for large problems, it will also hit the storage bottleneck.

This motivates a *memory-limited, single-pass streaming* algorithm; each received sample is examined once and then discarded. Furthermore, we only have a $O(p)$ memory budget. In particular, this means that algorithms cannot form the full covariance matrix. Starting with the algorithm in [OK85], many algorithms have been proposed to deal with this scenario. For a long time only consistency results (e.g. [OK85]) were known for this class of algorithms, with the actual convergence properties and sample complexity remaining out of reach (cf. [MCJ13], references and discussion therein).

Contributions: My 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. The algorithm uses only $O(p)$ memory and performs a single pass over the data. It is easily parallelizable, and our multicore implementation with nearly linear scalability can be found in [Mit14]. Recently, we generalized our results working in the regime where an overwhelming number of sample entries are erased (missing). A preprint of our work can be found in [MCJ14].

Future Directions: The machine learning community has worked on a number of improvements and extensions to this problem since we published our results [BDF13, Sha14, HP14]. The existing literature depends on the existence of an eigen-gap in the spectrum. In practice, this is not guaranteed and in some scalings all gaps diminish. There is consensus that understanding this regime is important, but there is no successful analysis. Related to this is the more systematic study of the trade-offs between the memory budget, accuracy and robustness.

Computationally, our algorithm is efficient and parallelizable with the exception of the orthonormalization step, which is relegated to a single thread. The literature contains some work on distributed QR factorizations (e.g. [ACD$^+$10]). The study of parallel – ideally non-blocking – algorithms for large scale orthonormalization is important in this context and a very interesting research direction in general.

Large Graph Analytics

Graphs make for a powerful combinatorial tool, when it comes to modeling real world interactions. The field of network sciences attracts a diverse group of disciplines: from sociologists studying the spread of obesity [CF07], to engineers analyzing the internet topology [FFF99] and computer scientists describing small-world phenomena [Kle00]. The study of graph properties (diameter, conductance, clustering) is fundamental for all of these applications. Existing networks are already too large for some established algorithms. Significant work is being put into designing large scale algorithms [HP96, GBSW10]. A number of them ([PSL90, DFK$^+$04, SS11]) draw tools from spectral graph theory [Chu97]. The reason is that the spectrum of a graph codifies a number of important properties like connectivity and mixing time. The eigendecomposition can be approximated reasonably fast even for large graphs and it provides a succinct approximation of an otherwise complex graph. We put some of these ideas to use in our work on large graphs.

A. Densest k-Subgraphs on MapReduce

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 [Kho06]. The best known result is a $O(n^{1/4+\epsilon})$ approximation ([BCC$^+$10]). 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.

Contributions: In [PMDC14] we do that. Using a low-rank approximation for the adjacency matrix, and leveraging recent combinatorial methods for quadratic optimization over low-rank spaces ([KL10]) we are able to get an efficient solver for this problem. Furthermore, the spectral decomposition in the first step, provides us with a *data-dependent* bound on the quality of the achieved 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. We implemented our algorithm on MapReduce [DG08] and solved for graphs with billions of edges on Amazon Web Services' EC2 using a cluster of 100 machines. The code is available on our software repository [MP14].

B. Fast PageRank Approximations on Graph Engines

PageRank ([PBMW99]) was introduced by Page et al. in order to "bring order to the web." Intuitively, web pages are the nodes of the web graph with hyperlinks being its directed edges. The PageRank vector is a probability distribution assigning more mass to important/central web pages. PageRank's intuitive nature and robustness to noise/manipulation have led to its application in an impressive number of domains (cf. review paper [Gle14]). When it comes to search engines, one just needs to find the top few dozen web pages for a query. This generalizes to most applications: we are mostly interested in the "heavy hitters," so we look for a good k-sized set. We quantify goodness as the amount of actual PageRank mass captured in the k-set.

Motivated by discussions with people in the industry, we work in the context of *graph engines* ([LGK$^+$10, Ave11, XGFS13]). They partition the graph across many machines and expose to the programmer a simple yet powerful API, the *vertex program*. The programmer gives simple instructions that will be executed by every vertex of the graph until convergence. We choose to work on GraphLab [LGK$^+$10] as it has proven to be the fastest among distributed engines (cf. [SSP$^+$]) for this kind of task.

Contributions: 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 (cf. our paper [MBDC14] for details). This change reduces the communication requirements of the algorithm significantly. As a side-effect, the random walks on the graph are not independent anymore, posing an analytical challenge. We nonetheless demonstrated experimentally and analytically that our method gives a good approximation with a 7x to 10x of speed improvement over the state of the art (vanilla GraphLab). The maintainers of the open source branch of GraphLab have requested our modifications for inclusion in the project.

Future Directions: This work opens many possibilities for future research. The gains we get in this case by reducing network communication (a very common bottleneck) suggest that randomized synchronization should be beneficial in many more graph analytics tasks. Single source shortest paths [Ram97], triangle counting [SW05] and k-profiles [HLN+14] are of particular interest to us.

Another important future direction is the connection between PageRank to general problems of ranking. Recent work in [OS14] suggests a very intuitive connection between a class of ranking problems (the mixed multinomial logit model) and random walks on a specific graph. This implies that an efficient algorithm for approximating PageRank can be used to learning mixed rankings directly. This connection is characteristic of a trend. Graphs are appearing as tools rather than models for real networks. Engineers in the industry often use a *transactional characterization*: a recommender system creates and adjusts the edges between different alternatives as it receives the users' preferences from the front-end. As graph engines are already deployed by many companies, efficient ranking algorithms on that framework is a timely and potentially very impactful problem.

Information Theoretic Bounds

The machine learning community uses tools like oracle analysis and Cramér-Rao bounds (e.g. [HOX14]) to reason about fundamental (or "minimax") limits. Information theory has some invaluable tools to offer, that are often overlooked. One result used occasionally is Fano's inequality [AWBR09]. It 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.

Contributions: Our work in [MV10] introduces a 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 many of the same tools to provide tight achievability and strong converse results for the task of learning a number of rankings, jointly from many users' pairwise preferences.

Future Directions: With a few exceptions, (e.g. [HOX14]), there are not many converse results for ranking. Our work provides tight bounds, but an extension to more general, noisy, mixed ranking models (e.g. the model in [OS14]) would provide an invaluable fundamental limit to compare algorithms against. More generally, our method could be applied to strengthen a number of converse results in the literature [ASZ08, AWBR09, CD13].

References

[ACD+10] E. Agullo, C. Coti, J. Dongarra, T. Herault, and J. Langem. Qr factorization of tall and skinny matrices in a grid computing environment. In *Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pages 1–11. IEEE, 2010.

[ASZ08] Shuchin Aeron, Venkatesh Saligrama, and Manqi Zhao. Information theoretic bounds for compressed sensing. *arXiv preprint arXiv:0804.3439*, 2008.

[Ave11] C. Avery. Giraph: Large-scale graph processing infrastruction on hadoop. *Proc. of Hadoop Summit. Santa Clara, USA:[sn]*, 2011.

[AWBR09] Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

[BBAP05] Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of Probability*, pages 1643–1697, 2005.

[BCC+10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an o (n 1/4) approximation for densest k-subgraph. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 201–210. ACM, 2010.

[BDF13] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental pca. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.

[CD13] Emmanuel J Candes and Mark A Davenport. How well can we estimate a sparse vector? *Applied and Computational Harmonic Analysis*, 34(2):317–323, 2013.

[CF07] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[Chu97] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[DFK+04] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering large graphs via the singular value decomposition. *Machine learning*, 56(1-3):9–33, 2004.

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[GBSW10] A. Gubichev, S. Bedathur, S. Seufert, and Gerhard Weikum. Fast and accurate estimation of shortest paths in large graphs. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 499–508. ACM, 2010.

[Gle14] David F Gleich. Pagerank beyond the web. *arXiv preprint arXiv:1407.5107*, 2014.

[HLN+14] Hao Huang, Nati Linial, Humberto Naves, Yuval Peled, and Benny Sudakov. On the 3-local profiles of graphs. *Journal of Graph Theory*, 76(3):236–248, 2014.

[HOX14] Bruce Hajek, Sewoong Oh, and Jiaming Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems*, pages 1475–1483, 2014.

[HP96] STEVEN Homer and M Peinado. Experiments with polynomial-time clique approximation algorithms on very large graphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26:147–168, 1996.

[HP14] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems*, pages 2861–2869, 2014.

[Jol05] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[Kho06] Subhash Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

[KL10] George N Karystinos and Athanasios P Liavas. Efficient computation of the binary vector that maximizes a rank-deficient quadratic form. *Information Theory, IEEE Transactions on*, 56(7):3581–3593, 2010.

[Kle00] Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845–845, 2000.

[LGK+10] Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M Hellerstein. Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv:1006.4990*, 2010.

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

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

[MCJ14] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Streaming PCA with Many Missing Entries. *Preprint*, 2014.

[MGCV11] I. Mitliagkas, A. Gopalan, C. Caramanis, and S. 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.

[MP14] Ioannis Mitliagkas and Dimitris Papailiopoulos. Spannogram library. `https://github.com/migish/spannogram`, 2014. Accessed: 2014-06-21.

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

[OK85] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.

[OS14] Sewoong Oh and Devavrat Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.

[PBMW99] L. Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[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.

[PSL90] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.

[Ram97] Rajeev Raman. Recent results on the single-source shortest paths problem. *ACM SIGACT News*, 28(2):81–87, 1997.

[Sha14] Ohad Shamir. A stochastic pca algorithm with an exponential convergence rate. *arXiv preprint arXiv:1409.2848*, 2014.

[SS11] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM JOC*, 40(6):1913–1926, 2011.

[SSP+] Nadathur Satish, Narayanan Sundaram, Mostofa Ali Patwary, Jiwon Seo, Jongsoo Park, M Amber Hassaan, Shubho Sengupta, Zhaoming Yin, and Pradeep Dubey. Navigating the maze of graph analytics frameworks using massive graph datasets.

[SW05] Thomas Schank and Dorothea Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. In *Experimental and Efficient Algorithms*, pages 606–609. Springer, 2005.

[XCS10] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[XGFS13] Reynold S Xin, Joseph E Gonzalez, Michael J Franklin, and Ion Stoica. Graphx: A resilient distributed graph system on spark. In *First International Workshop on Graph Data Management Experiences and Systems*, page 2. ACM, 2013.