Research Statement

Lev Reyzin

November 2017

My research lies in computational learning theory, as well as in the broader fields of theoretical computer science and machine learning. A large fraction of my work focuses on what is called interactive machine learning. This notion captures learning algorithms that can somehow interact with their source of data, thereby affecting which data they learn from.

What follows is a brief summary of my past and ongoing work, where I highlight several results representing the diverse fields I work in. All of the diagrams and theorems appearing explicitly herein are stated (or restated) from my papers. All references to my work are typeset in boldface font to distinguish my results from the work of others.

1 Computational Learning Theory

Much of computational learning theory focuses on exploring and categorizing the learnability of various function classes under different learning models. I begin mainly by discussing some of my older work on learning graphs, circuits, and automata before moving on to more recent contributions.

1.1 Learning languages and automata

Deterministic finite state automata (DFA), which recognize exactly the regular languages, are one of the most fundamental objects in all of computer science, but their learnability has not been fully understood. Membership queries allow a learner to pick a string x and return whether or not x is in the target language. While it is known that automata cannot be efficiently exactly learned from membership information alone, Angluin [Ang87] showed that DFA are learnable with membership queries if equivalence queries are also added.

But, among other results, my colleagues and I showed that access to random bits "sprinkled" on the states of a hidden random automaton allows for efficient learning with membership queries alone (cf. Freund et a. [FKR$^+$97]). Our other results extend to all c-concentrating automata.

Theorem 1 ([**ABBDR09**])**.** *A finite automaton with n states, a random transition function and a random labeling can, with high probability, be learned using $O(n \log(n))$ label queries.*

On the other hand, DFA appear harder to learn in the PAC model, where a learner is given access to random labeled examples from a target distribution and must predict the labels of future data points from the same distribution. PAC learning DFA is "cryptographically hard" [KV94]; moreover, they are also known to be unlearnable in the "statistical query" model of learning (SQ), which is a natural restriction of the PAC model [Kea98]. In efficient SQ learning, the algorithm does not interact with data, but rather asks efficiently computable queries (which are functions of

example features and their labels according to the target concept) to an SQ oracle that returns the expectation of the query function on the data distribution within a polynomial "tolerance," which is also a parameter. This model is made to capture algorithms that can operate by only looking at "statistical properties" of the data, without relying on any particular examples themselves.

My coauthors and I were able to strengthen the latter results, proving that even random DFA (as well as random DNF expressions and random decision trees) remain polynomial-time unlearnable with statistical queries. This was proven by choosing a target data distribution to w.h.p. force the random automaton to compute parity functions, which have high "statistical query dimension," a known impediment for statistical query learning [BFJ$^+$94].

Theorem 2 ([**AEKR10**]). *No algorithm can weakly learn random deterministic finite acceptors with n states with respect to an arbitrary distribution on strings of length at most* $\Theta(log^3 n)$ *using a polynomial number of statistical queries.*

Hence, the hope is to be able to learn random automata under the *uniform* distribution on inputs. There has been recent progress on this by Angluin and Chen [AC15], but their algorithms require additional state information. My students and I have been devising new methods for this problem; so far, we have preliminary results that employ and derive Discrete Fourier Analysis techniques to get PAC algorithms that do not need any additional state information to learn DFAs [**FR17b**].

1.2 Learning Sparse Parities with Noise

Another related fundamental problem that I've tackled is called "noisy parity." Parity functions play a central role in learning theory, as it gives perhaps the only natural example of a class of functions that is known to be PAC learnable, but not SQ learnable (see Section 1.1). The noisy parity problem asks to PAC learn parity functions under random noise on the labels, i.e. each label has an independent probability of η of being flipped. This model of learning under randomly corrupted labels is also called PAC learning under white-label noise, or "noisy PAC" for short [AL87].

In a breakthrough result that separated noisy PAC from SQ as learning complexity classes, Blum et al. [BKW03] gave a better than brute-force algorithm for learning noisy parities that ran in time $2^{O(n/\log n)}$. However, in the case where the parity functions are restricted to be r-sparse, i.e. concentrated on at most r variables, no algorithm faster than the brute force $O(n^r)$ was known before my work with Elena Grigorescu and Santosh Vempala.

Theorem 3 ([**GRV11**]). *The class of r-parity functions is learnable in the noisy PAC model in time* $\text{poly}\left(\log(1/\delta), 1/(1-2\eta)\right) n^{(1+2\eta+o(1))r/2}$, *using* $m = \omega\left(\frac{r\log(n/\delta)}{(1-2\eta)^2}\right)$ *samples.*

This bound begins at $O(n^{r/2})$ for $\eta = 0$, but degrades to the brute-force $O(n^r)$ as η approaches $1/2$. Our result applied the "nearest neighbor" approach informally suggested by Hopper and Blum [HB01] to the noisy parity problem. This nearest neighbor approach was then improved by Valiant [Val15] in his breakthrough work that gave an algorithm for learning noisy parity whose running time was still exponential, $\approx O(n^{.8r})$, but whose exponent did not degrade as $\eta \to 1/2$.

1.3 Learning hidden graphs and evolutionary trees

Another important area in query learning uses models originating in computational biology, especially when the target concepts to be learned are graphs.

Perhaps the most famous of these is the problem of evolutionary tree reconstruction. A costly experiment can uncover the genetic distance between any two species, and from evolutionary biology, we know that these distances are "tree-realizable," i.e. embeddable into a tree without distortion – the goal then is to reconstruct the species' evolutionary tree using as few experiments as possible. The most natural approach is to recursively find what are called "long paths" in the evolutionary tree, stitching them bach onto the parent branch when coming out of the recursion. This `Longest Path` approach [CR89] was long thought to run in time $O(dn \log_d n)$ for trees of maximum degree d, which is known to be optimal from a much more complicated algorithm due to Hein [Hei89], Nikhil Srivastava and I, however, found a counterexample; namely, we discovered an infinite family of trees on which the longest path approach requires $\Theta(d^{1/2}n^{3/2})$ queries. We were then able to prove that this is, in fact, the tight worst-case bound for the algorithm, correcting a long-held inaccuracy in the evolutionary tree reconstruction literature.

Theorem 4 ([**RS07b**]). *The worst-case running time of the* `Longest Path` *algorithm for learning n-node evolutionary trees with all node degrees bounded by d is $\Theta(d^{1/2}n^{3/2})$.*

In another work on graph learning, Nikhil Srivastava and I analyzed the theoretical properties of a large variety of queries for the problems of graph learning and verification [**RS07a**]. There, we proved many new results for a variety of graph query models.

Most surprisingly, perhaps, we were able to show that any graph is verifiable to any failure rate bounded by $\epsilon > 0$ using $O(\log(1/\epsilon))$ "edge-counting" queries, independent the size of the input graph. Edge-counting queries receive a subset of vertices of the graph and return the number of edges in the subgraph induced by the given vertices. These queries capture the "additive model" in bioinformatics, which was inspired by computational problems in Multiplex PCR, a widely employed technique for quickly detecting deletions and duplications in a gene [BGK05].

The main technical work of the proof relied on extending Freivalds's "matrix fingerprinting" results [Fre77] to settings where the fingerprints are computed by multiplying the target matrix by the same random vector from both the left and the right. We were able to obtain the following, perhaps surprising, theorem, which has immediate applications to graph verification when interpreting the vector v as "selecting" vertices to participate in the query: $v^T Av$ is the edge-count of the subgraph induced by a query vector v from a graph whose adjacency matrix is A.

Theorem 5 ([**RS07a**]). *Let A and B be $n \times n$ symmetric matrices over a field such that $A \neq B$, then for v chosen uniformly at random from $v \in \{0,1\}^n$, $\Pr[v^T Av \neq v^T Bv] \geq 1/4$*

1.4 Learning circuits by injecting values

In a different line of research, my coauthors and I greatly extended the understanding of Value Injection Queries (VIQs) – a query model for learning circuits that was motivated by problems in discovering gene regulatory networks [AACW09]. Traditional circuit-learning models focus on the complexity of learning circuits by manipulating their inputs, but few classes of circuits are learnable in those models. VIQs give the learner the power to "inject values" into the gates of the hidden circuit and leave others "free," but only observe the value on the output gate. The connections among the gates are also unknown to the learner in the VIQ model of learning circuits.

While the complexity of deterministic learning boolean circuits with VIQs was known, we showed that the results become surprisingly different for analog circuits, or simply when large finite alphabet sizes are considered. Here, the "shortcut width" of a circuit plays a role, and we have a lower bound nearly meeting the bound in the theorem below.

Theorem 6 ([**AACR08**]). *The class of n-gate circuits with "shortcut width" bound b, fan-in bound k, and alphabet size s can be learned using $(ns)^{O(k+b)}$ queries.*

This theorem, like most positive results in the literature relies on analyzing "test paths," which are basically sequences of free gates from a given gate to the output. Variants of "test path lemmas" for circuits basically say that the full behavior of a circuit could be inferred from the circuit's behavior only on test paths. These lemmas limit the set of queries a learning algorithm should consider, thereby simplifying the job of finding efficient learning algorithms.

Yet, certain types of circuits are resilient to the use of test paths. In a separate work, we also considered the case of learning probabilistic circuits (also known as Bayesian networks) in the VIQ model [**AAC$^+$09**]. Here we showed that test paths "attenuate" exponentially in probabilistic circuits, and utterly fail the test path lemma when the alphabet size is > 2.

Then, my student Jeremy Kun and I extended the VIQ model to also include quantum circuits, and we gave some preliminary results. Unsurprisingly, we were able to prove that quantum circuits will also fail the test path lemma.

Theorem 7 ([**KR15**]). *There is a circuit on which every VIQ leaving a path free makes the last output qubit uniformly random, yet with no VIQ the last output qubit is deterministic.*

1.5 Learning social networks

Dana Angluin, James Aspnes, and I adapted the "value injection query" model to learning independent cascade social networks [**AAR10b**]. VIQs on social networks correspond to the natural operations of activating and suppressing agents in the network. We devised an optimal algorithm for learning social networks with VIQs. My more recent work on social networks focused on the ability of a passive learner to infer a social network by making observations.

In [**AAR10a**] we tackled the problem of a learner inferring the most likely structure connecting a population after it observes how diseases, or other outbreaks, spread through that population. There, we gave algorithms for reconstructing the most likely social networks from such data. This work had some interesting subsequent applications, including on product new techniques for generating semantic maps in the linguistics community [RKM13].

In [**FHR16**], my students and I worked on a related problem of learning social networks from simply observing voting data. The idea is if a set of agents are repeatedly, publicly voting, while having their votes influenced by a social network, then it may be possible, after enough observations of the votes, to recover the social network structure itself. We considered two similar models of how social networks may influence voting: an edge-centric and a vertex-centric model. We showed that the resulting problem complexity and learned network structures could be quite different between the two, illustrating that slight variations in

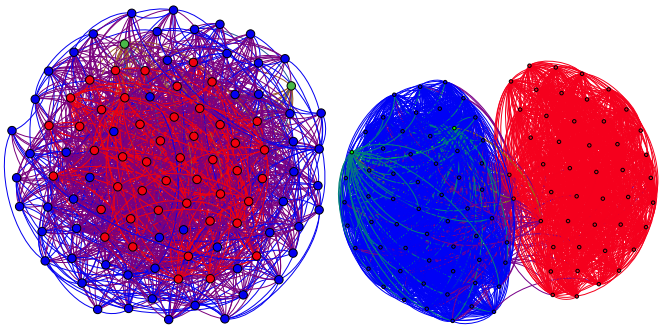

Figure 1: Left: the network learned by a vertex-centric algorithm, and right: the network learned by an edge-centric algorithm, when run senate votes from the 2013-2014 term. Nodes corresponding to Democrats, Republicans, and Independents are colored blue, red, and green, respectively.

modeling may drastically affect the structure of learnt networks. Our conclusion was that significant care needs to be taken in choosing the right model in order for such results to be meaningful.

My student Ben Fish and I have also continued the study this and related models from a pure learning-theory perspective as a variant of multiple instance learning called "learning from label proportions" [**FR17a**].

1.6 Interactive clustering

My student Ádám Lelkes and I worked on the interactive clustering model proposed by Balcan and Blum [BB08]. This new clustering model was one of the first to model the possibility of a human being assisting a clustering algorithm find a good clustering. In this query model, a person can ask the algorithm to "split" an impure cluster or to 'merge" two clusters if their union should be a pure (or a subset of a pure) cluster in the target clustering.

In addition to finding broad classes of linear functions which could be clustered interactively, we gave the first lower bounds for this model. Using ideas from the cryptographic lower bounds Kearns an Valiant developed for PAC learning [KV94], we were also able to prove the first lower bounds in this model. This was not so straightforward an application of their results since, unlike in PAC, there is no "future data" which could be used to test whether a particular cryptographic primitive is broken by the algorithm.

Theorem 8 ([**LR15**])**.** *If there are $n^{\omega(1)}$-hard pseudorandom function families computable in logarithmic space, then polynomial-size DFAs are not efficiently learnable in the interactive clustering model.*

2 Machine learning: theory and practice

Another direction of my work focuses on problems driven by real-world concerns. Much this work is mathematical, but it carries a different flavor than my computational learning theory research, and some this work is empirical; most is both.

2.1 Bandit algorithms for computational advertising

An important problem for many internet companies is what content to display. Search engines receive context about their users and after displaying content, they get feedback on what the users clicked on, as well as some reward which they try to maximize. They get no explicit information, however, about how specific users would react if they were shown different content – partial feedback problems of this nature are called "bandit" problems. A good bandit algorithm must balance exploring new options and exploiting what has already been learned. In the contextual bandit problem, the algorithm uses the context information to compete with a large class of policies.

While I was at Yahoo! Research, my colleagues and I developed the first efficient high-probability optimal algorithm for the contextual bandit problem, which we called `EXP4.P` [**BLL$^+$11**]. We then continued the research by pursuing an idea introduced by Langford and Zhang [LZ07] and we gave an optimal algorithm (i.e. with $O(\sqrt{TK}\text{polylog}N)$ "regret") that could be made efficient given a cost-sensitive supervised learning oracle, essentially giving a reduction from finding an optimal and efficient bandit algorithm to the well-studied problem of cost-sensitive supervised PAC learning.

Theorem 9 ([**DHK**$^+$**11**]). *In a contextual bandit problem, let T be the number of timesteps, K the number of arms (we assume $K \leq T$), and $\mathcal{F} : X \to \{1, \ldots, K\}$ a class of functions with $N = |\mathcal{F}|$. Then, given a cost-sensitive supervised learning oracle to $\mathcal{F}$, an $\tilde{O}(\sqrt{TK \log N/\delta})$-regret algorithm exists that runs in time polynomial in T, K, $1/\delta$, and $\log N$, failing with probability at most $\delta > 0$.*

I note that this result was only a theoretical breakthrough. Our algorithm employed the cost-sensitive oracle to find separating hyperplanes for constructing a separation oracle to be used to run the Ellipsoid method for a convex optimization problem that we constructed. This algorithm was therefore only theoretically efficient and could not practically be implemented. However, our result was later improved by Agarwal et al., who found a much simpler algorithm with the same guarantees, which they were able to implement and achieve impressive experimental results [AHK$^+$14].

Satyen Kale, Robert Schapire, and I also generalized this problem for the case when multiple ads can be displayed to users at once [**KRS10**]. Then, my colleagues and I also proved the first theoretical guarantees, as well as generic lower bounds, for LinUCB a natural algorithm that works under a restricted "linear payoffs" setting.

Theorem 10 ([**CLRS11**]). *For T rounds, K, actions, and d dimensional feature vectors, the decomposed* `LinUCB` *algorithm has regret of $\tilde{O}(\sqrt{Td}\log^{3/2}(K/\delta))$ w.p. $1-\delta$. Moreover, when $d^2 \leq T$ every algorithm has worst-case expected regret of $\gamma\sqrt{Td}$ for some constant $\gamma > 0$.*

I note that not only did the algorithms arising from this research make exciting advances in bandit learning theory, some of these, notably our aforementioned `EXP4.P` algorithm, were tested experimentally and produced improved estimated click-through rates in simulations on datasets containing tens of millions of user visits to Yahoo! news [**BLL**$^+$**11**]. McMahan has argued that understanding exactly why `EXP4.P` worked so well within Yahoo!'s particular system is in itself an intriguing research challenge [McM11].

2.2 Boosting

Boosting [FS97] is a technique that effectively combines many weak predictors to get a strong machine learning algorithm – it actually has many nontrivial connections to the bandit algorithms presented in the previous section.

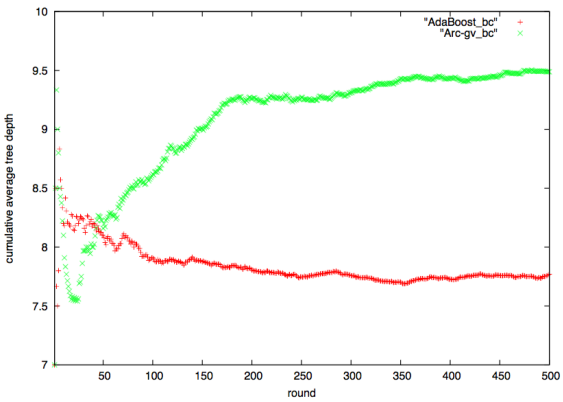

Figure 2: Cumulative average of decision tree depth for AdaBoost and arc-gv for the breast cancer set for 500 rounds of boosting.

Robert Schapire and I tackled the problem of reconciling theoretical predictions from the margins theory of boosting [SFBL98] with experimental data. Experiments by Leo Breiman put the margins theory for the effectiveness boosting into serious doubt [Bre99], but we showed how the margins theory explains his results and also discovered new phenomena. Our examination revealed a complex interplay between the choice of boosting algorithm, the margins it achieves on training examples, and the complexity of the weak learners it produces [**RS06**]. For instance, Figure 2 shows how the depth of decision trees produced by two boosting algorithms, AdaBoost and arc-gv, can vary significantly.

Our work has influenced subsequent margin bounds, e.g. the equilibrium margin [WSJ+11], and the design of new boosting algorithms that aim to optimize the margins distribution [ZZ14].

The idea of boosting is powerful in practice because weak predictors appear easier to explicitly design than strong ones. A plethora of theoretical and experimental work attempts to improve upon, explain, or re-examine the boosting procedure. However, little research has been done in trying to understand how to design a good weak learner itself. In **[Rey14]**, I gave conditions that a weak learner should satisfy for a boosting algorithm to work in practice. These conditions quickly implied that sparse parity functions could make for especially good weak learners (nicely relating to and employing my results in Section 1.2). My new simple approach was experimentally competitive with more sophisticated methods like pruned CART trees [BFOS84].

Finally, I have tackled problems where boosting ended up becoming a very useful tool, in particular in the area of feature-efficient prediction, a budgeted learning model. Here, the learner incurs a cost upon examining the features of an example, and the goal is to design an algorithm that minimizes prediction error subject to a constraint on the total cost of the features it examines. One of my results examines the relationship between the ability of an ensemble algorithm to achieve a good margins distribution on training data and its ability to make use of only a few features while making predictions on test data **[Rey11]**. Together with a few students, we have recently extended these results to explicitly incorporate feature costs in building boosting ensembles **[HPR15]** – our results are illustrated in Figure 3. This remains an active area of research interest for me, and I am currently working on extending support vector machines (SVM) to this problem setting.

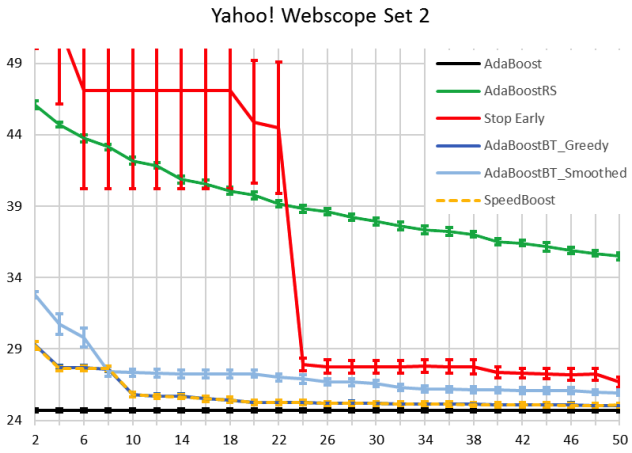

Figure 3: Our algorithms, in light blue and dark blue (exactly follows the yellow line), beat or match other approaches across experiments.

2.3 Overcoming inductive bias in active learning

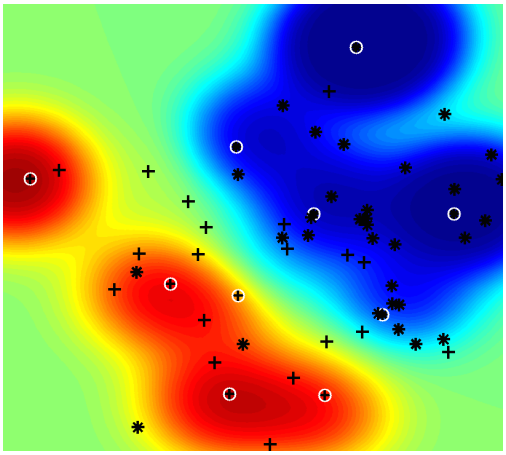

Figure 4: A "heatmap" of our algorithm's predictions. Red/blue toward +/*, resp.

In another line of work related to interactive learning, my colleagues and I recently took the view that active learning is actually mostly a covariate shift problem in disguise. A learning algorithm that uses a non-random strategy will make predictions using data that come from a different distribution than the one it will be tested on. Existing learning approaches also use their own predictions to produce a sampling strategy. These two observations can create a serious downward spiral – that a learning algorithm will be too confident in its own predictions and then make worse and worse sampling decisions. In the case of logistic regression, we took the max-entropy approach to give a learner pessimistic estimates of its own certainty. This led to a

provably good sampling and prediction strategy, which experimentally avoided the pitfalls of previous approaches [**LRZ14**].

The following theorem states that if we model correctly, our predictor's loss function upper bounds its expected generalization error motivated our approach and helps explain our experimental results in Figure 4, which clearly differ significantly from those of logistic regression..

Theorem 11 ([**LRZ14**])**.** *Assuming that the actual label distribution $P(y|x)$ is within the set $\tilde{\Xi}$, the full data entropy of the RBA predictor upper bounds its generalization loss:*

$$H_{\mathcal{D}}(Y|X) \triangleq \mathbb{E}_{P_{\mathcal{D}}(x)\hat{P}(y|x)}\left[-\log \hat{P}(Y|X)\right] \geq \mathbb{E}_{P_{\mathcal{D}}(x)P(y|x)}[-\log \hat{P}(Y|X)].$$

3 Theory

I have also worked on a variety of problems in theoretical computer science. As one can see from the work below, they are often related to my other research in machine learning and learning theory.

3.1 Planted Cliques and Statistical Algorithms

Some of my recent work has involved characterizing the power of statistical algorithms. Inspired by statistical queries in learning theory, my colleagues and I characterized a wide class of optimization algorithms that use statistical properties of their inputs. This characterization essentially captures many well-studied heuristics, including local search, MCMC, and simulated annealing. We showed that for a variety of optimization problems – notably a variant of the planted clique problem – statistical algorithms unconditionally require time exponential in their input parameters [**FGR$^+$17**]. Specifically, we proved the following about statistical algorithms, a class of algorithms we defined for optimization problems over distributions, that closely resembles statistical query algorithms from learning theory – we still have query functions and a tolerance parameter, but no labels.

Theorem 12 ([**FGR$^+$17**])**.** *No statistical algorithm exists for finding planted bi-clique distributions with planted clique sizes of $s = O(n^{1-\epsilon})$ (for any $\epsilon > 0$) that makes a polynomial number of queries of tolerance $\tau = \Omega(1/n)$.*

This was among the first general lower bounds explaining the lack of progress on the planted clique problem for clique sizes below $s = O(\sqrt{n})$, for which efficient algorithms are known [AKS98]. This result also elucidates some relationship between the noisy parity problem, which is known to have high statistical query dimension (see Section 1.2), and the planted clique problem, showing that both their difficulties may arise from a common source: high statistical dimension.

In a related work, Shmuel Friedland, Sam Cole and I are have been simplifying and generalizing statistical algorithms for a related problem called "planted clustering," where a clique partition is planted in a random graph. Our main result gives the simplest algorithm yet for recovering planted partitions with partition sizes $\Omega(n^{1/2})$ [**CFR15**]. Together with students, I am currently trying to also develop tight statistical lower bounds for this planted clustering variant, which would begin to resolve an open question relating its complexity to that of planted clique.

3.2 Computational Complexity of MapReduce

In a recent theoretical pursuit to understand networked computation, my co-authors and I began a study of the popular MapReduce from a computational complexity standpoint. In the formal

model, roughly, data is split among a sub-linear number of machines, such that each machine is restricted to seeing a sub-linear portion of the input. In **[FKL$^+$14]**, we proved the first hierarchy theorems for the MapReduce complexity class.

A language L is said to be in MRC$[f(n), g(n)]$, informally, if $O(n^{1-\epsilon})$ machines each space bounded by $O(n^{1-\epsilon})$, can together decide L using at most $f(n)$ global communication steps and at most $g(n)$ time-steps bounding computation steps per round. Our "MRC hierarchy" theorem, which also conditionally resolves a conjecture about TISP classes (see [For00]) from classical complexity, can then be stated as follows.

Theorem 13 (**[FKL$^+$14]**). *Assuming the Exponential Time Hypothesis (ETH), for every α, β there exist $\mu > \alpha$ and $\nu > \beta$ such that* $\text{MRC}[n^\alpha, n^\beta] \subsetneq \text{MRC}[n^\mu, n^\beta]$ *and* $\text{MRC}[n^\alpha, n^\beta] \subsetneq \text{MRC}[n^\alpha, n^\nu]$.

We also showed that in a constant number of rounds, using a MapReduce protocol, one can solve any problem the lies in sub-logarithmic space. I note that because our proof is constructive, it gives a black-box way to MapReduce any language requiring sub-logarithmic space, which contains a strictly larger set of languages than simply the regular languages [Sze94].

Theorem 14 (**[FKL$^+$14]**). *Let* $\text{MRC}^0 = \bigcup_{k \in \mathcal{N}} \text{MRC}[1, n^k]$, *then* $\text{SPACE}(o(\log n)) \subseteq \text{MRC}^0$.

In the era of "big data" understanding the foundations of what is possible in a distributed environment is an important research area that I am continuing to pursue, e.g. in **[CR15]**.

3.3 Clustering and coloring stability

One current trend in clustering is to circumvent the NP-hardness of optimizing various objective functions by examining which properties of the data, if satisfied, allow for polynomial time algorithms. One such assumption is called perturbation resilience, which states that the optimal clustering doesn't change even when the distances among the data points are perturbed by certain factors [BL10], cf. Ackerman and Ben-David [AB09].

In a recent work **[Rey12]**, I carefully analyzed this data resilience assumption and showed that the choice of resilience parameter can drastically affect certain clustering problems. I was able to show that that even with a significant amount of resilience, the problem remains NP-hard. Then, Shalev Ben-David and I showed that with some added resilience, the data begins to have very strong structural properties, allowing only for a narrow range of parameter values between where the problems are NP-hard and where they are trivial.

Theorem 15 (**[BR14]**). *Every $(2 + \sqrt{3})$-resilient clustering instance has the "strict separation" property – namely, every point is closer to every other point in its own cluster than to any point in another cluster.*

In a related line of work, My student Jeremy Kun and I have extended the notion of "stability" to any constraint satisfaction problem (CSP), in particular analyzing graph coloring and boolean satisfiability. For coloring, r-resilience implies that a graph remains colorable even after the addition of any r edges. So, a 1-resilient 3-colorable graph is not only 3-colorable, but remains so after one edge is added anywhere in the graph. After taking proving the hardness of resilient-k-SAT for any non-trivial setting of the parameters, we employed a useful reduction "gadget" relating 3-SAT to coloring problems that we invented in a previous work **[KPR13]** in order to prove our main result.

Theorem 16 (**[KR14]**). *It is NP-Hard to* 3*-color* 1*-resilient* 3*-coloring instances.*

On the other hand, we know that it is easy to 3 color 3-resilient 3-colorable graphs, leaving the problem of 2-resilient 3-colorability tantalizingly open.

Finally, Will Perkins and I are also pursuing research on another notion of stability in graphs, where stability is defined w.r.t. a graph's resistance to spreading cascades among its vertices **[PR13]**.

References

[AAC+09] Dana Angluin, James Aspnes, Jiang Chen, David Eisenstat, and Lev Reyzin. Learning acyclic probabilistic circuits using test paths. *Journal of Machine Learning Research*, 10:1881–1911, 2009. Previous version in *COLT* (2008).

[AACR08] Dana Angluin, James Aspnes, Jiang Chen, and Lev Reyzin. Learning large-alphabet and analog circuits with value injection queries. *Machine Learning*, 72(1-2):113–138, 2008. Previous version in *COLT* (2007).

[AACW09] Dana Angluin, James Aspnes, Jiang Chen, and Yinghua Wu. Learning a circuit by injecting values. *J. Comput. Syst. Sci.*, 75(1):60–77, 2009.

[AAR10a] Dana Angluin, James Aspnes, and Lev Reyzin. Inferring social networks from outbreaks. In *ALT*, pages 104–118, 2010.

[AAR10b] Dana Angluin, James Aspnes, and Lev Reyzin. Optimally learning social networks with activations and suppressions. *Theor. Comput. Sci.*, 411(29-30):2729–2740, 2010. Previous version in *ALT* (2008).

[AB09] Margareta Ackerman and Shai Ben-David. Clusterability: A theoretical study. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, pages 1–8, 2009.

[ABBDR09] Dana Angluin, Leonor Becerra-Bonache, Adrian Horia Dediu, and Lev Reyzin. Learning finite automata using label queries. In *ALT*, pages 171–185, 2009.

[AC15] Dana Angluin and Dongqu Chen. Learning a random DFA from uniform strings and state information. In *Algorithmic Learning Theory - 26th International Conference, ALT 2015, Banff, AB, Canada, October 4-6, 2015, Proceedings*, pages 119–133, 2015.

[AEKR10] Dana Angluin, David Eisenstat, Leonid Kontorovich, and Lev Reyzin. Lower bounds on learning random structures with statistical queries. In *ALT*, pages 194–208, 2010.

[AHK+14] Alekh Agarwal, Daniel J. Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1638–1646, 2014.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Struct. Algorithms*, 13(3-4):457–466, 1998.

[AL87] Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.

[BB08] Maria-Florina Balcan and Avrim Blum. Clustering with interactive feedback. In *Algorithmic Learning Theory, 19th International Conference, ALT 2008, Budapest, Hungary, October 13-16, 2008. Proceedings*, pages 316–328, 2008.

[BFJ+94] Avrim Blum, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 253–262, 1994.

[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[BGK05] Mathilde Bouvel, Vladimir Grebinski, and Gregory Kucherov. Combinatorial search on graphs motivated by bioinformatics applications: A brief survey. In *Graph-Theoretic Concepts in Computer Science, 31st International Workshop, WG 2005, Metz, France, June 23-25, 2005, Revised Selected Papers*, pages 16–27, 2005.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BL10] Yonatan Bilu and Nathan Linial. Are stable instances easy? In *ICS*, pages 332 – 341, 2010.

[BLL+11] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. *AISTATS*, pages 19–26, 2011.

[BR14] Shalev Ben-David and Lev Reyzin. Data stability in clustering: A closer look. *Theor. Comput. Sci.*, 558:51–61, 2014.

[Bre99] Leo Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1517, 1999.

[CFR15] Sam Cole, Shmuel Friedland, and Lev Reyzin. A simple spectral algorithm for recovering planted partitions. *CoRR*, abs/1503.00423, 2015.

[CLRS11] Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. *AISTATS*, pages 208–214, 2011.

[CR89] Joseph C. Culberson and Piotr Rudnicki. A fast algorithm for constructing trees from distance matrices. *Inf. Process. Lett.*, 30(4):215–220, 1989.

[CR15] Jeff Cooper and Lev Reyzin. Improved algorithms for distributed boosting. Unpublished manuscript, 2015.

[DHK+11] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *UAI*, pages 169–178, 2011.

[FGR+17] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Srinivas Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *J. ACM*, 64(2):8:1–8:37, 2017.

[FHR16] Benjamin Fish, Yi Huang, and Lev Reyzin. Recovering social networks by observing votes. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2016, Singapore, May 9-13, 2016*, pages 376–384, 2016.

[FKL+14] Benjamin Fish, Jeremy Kun, Ádám Dániel Lelkes, Lev Reyzin, and Gyorgy Turan. On the computational complexity of mapreduce, 2014.

[FKR+97] Yoav Freund, Michael J. Kearns, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. Efficient learning of typical finite automata from random walks. *Inf. Comput.*, 138(1):23–48, 1997.

[For00] Lance Fortnow. Time-space tradeoffs for satisfiability. *J. Comput. Syst. Sci.*, 60(2):337–353, 2000.

[FR17a] Benjamin Fish and Lev Reyzin. On the complexity of learning from label proportions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1675–1681, 2017.

[FR17b] Benjamin Fish and Lev Reyzin. Open problem: Meeting times for learning random automata. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 8–11, 2017.

[Fre77] Rusins Freivalds. Probabilistic machines can use less running time. In *IFIP Congress*, pages 839–842, 1977.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[GRV11] Elena Grigorescu, Lev Reyzin, and Santosh Vempala. On noise-tolerant learning of sparse parities and related problems. In *ALT*, 2011.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 52–66, 2001.

[Hei89] Jotun Hein. A tree reconstruction method that is economical in the number of pairwise comparisons used. *Molecular biology and evolution*, 6(6):669–684, 1989.

[HPR15] Yi Huang, Brian Powers, and Lev Reyzin. Training-time optimization of a budgeted booster. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3583–3589, 2015.

[Kea98] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.

[KPR13] Jeremy Kun, Brian Powers, and Lev Reyzin. Anti-coordination games and stable graph colorings. In *Algorithmic Game Theory - 6th International Symposium, SAGT 2013, Aachen, Germany, October 21-23, 2013. Proceedings*, pages 122–133, 2013.

[KR14] Jeremy Kun and Lev Reyzin. On coloring resilient graphs. In *MFCS*, pages 517–528, 2014.

[KR15] Jeremy Kun and Lev Reyzin. Open problem: Learning quantum circuits with queries. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 1767–1769, 2015.

[KRS10] Satyen Kale, Lev Reyzin, and Robert Schapire. Non-stochastic bandit slate problems. In *NIPS*, pages 1045–1053, 2010.

[KV94] Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

[LR15] Ádám Dániel Lelkes and Lev Reyzin. Interactive clustering of linear classes and cryptographic lower bounds. In *Algorithmic Learning Theory - 26th International Conference, ALT 2015, Banff, AB, Canada, October 4-6, 2015, Proceedings*, pages 165–176, 2015.

[LRZ14] Anqi Liu, Lev Reyzin, and Brian Ziebart. Pessimistic active learning using robust bias-aware prediction. manuscript, 2014.

[LZ07] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, 2007.

[McM11] H. Brendan McMahan. Discussion of "contextual bandit algorithms with supervised learning guarantees". In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 27–28, 2011.

[PR13] Will Perkins and Lev Reyzin. On the resilience of bipartite networks. *CoRR*, abs/1306.5720, 2013.

[Rey11] Lev Reyzin. Boosting on a budget: Sampling for feature-efficient prediction. In *ICML*, pages 529–536. ACM, 2011.

[Rey12] Lev Reyzin. Data stability in clustering: A closer look. In *ALT*, pages 184–198, 2012.

[Rey14] Lev Reyzin. On boosting sparse parities. In *AAAI*, pages 2055–2061, 2014.

[RKM13] Terry Regier, Naveen Khetarpal, and Asifa Majid. Inferring semantic maps. *Linguistic Typology*, 17(1):89–105, 2013.

[RS06] Lev Reyzin and Robert E. Schapire. How boosting the margin can also boost classifier complexity. In *ICML*, pages 753–760, 2006.

[RS07a] Lev Reyzin and Nikhil Srivastava. Learning and verifying graphs using queries with a focus on edge counting. In *ALT*, pages 285–297, 2007.

[RS07b] Lev Reyzin and Nikhil Srivastava. On the longest path algorithm for reconstructing trees from distance matrices. *Inf. Process. Lett.*, 101(3):98–100, 2007.

[SFBL98] Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pages 1651–1686, 1998.

[Sze94] Andrzej Szepietowski. *Turing Machines with Sublogarithmic Space*, volume 843 of *Lecture Notes in Computer Science*. Springer, 1994.

[Val15] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2):13, 2015.

[WSJ+11] Liwei Wang, Masashi Sugiyama, Zhaoxiang Jing, Cheng Yang, Zhi-Hua Zhou, and Jufu Feng. A refined margin analysis for boosting algorithms via equilibrium margin. *Journal of Machine Learning Research*, 12:1835–1863, 2011.

[ZZ14] Teng Zhang and Zhi-Hua Zhou. Large margin distribution machine. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 313–322, 2014.