Fast agnostic classification

Shiva Kaul
skkaul@cs.cmu.edu
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

A learning algorithm is agnostic if it doesn't presume a perfect model of how input data produce output data. Such algorithms are difficult to design, even for the basic task of classifying data as well as the best linear separator. This has led to a persistent rift between practice and theory: popular algorithms, such as SVMs and logistic regression, are susceptible to noise; provable agnostic algorithms involve brute-force sampling (which uses too much time) or fitting polynomials (which uses too much data.)

We recently introduced a new classification algorithm, KG, which is both practical and agnostic. It revisits basic elements of learning: 1. What functions should the algorithm fit? Smooth lists of halfspaces are a novel generalization of halfspaces. They are more flexible than halfspaces, but do not require more data to train in the worst case. 2. How should the algorithm fit such a function to the data? The algorithm involves 'immutable' iterations which are fundamentally different than update rules such as gradient descent, multiplicative weights, or perceptrons. KG achieves promising experimental performance for both natural and artificial problems.

We seek to deepen our theoretical understanding of the algorithm and expand its practical applications. The main question we shall answer is: when is KG provably fast? It eventually converges to the correct solution for a wide variety of input distributions. However, these intersect with a litany of hardness results, so restricting the input distribution seems necessary. Based on experimental evidence and the mechanics of the algorithm, we believe it is possible the algorithm runs in polynomial time when the inputs are normally distributed. If so, this algorithm would solve a notorious problem in computer science: learning logarithmically-sparse parities with noise. This would resolve a variety of challenges in learning theory, such as learning DNFs (encountered in 1984 by Valiant) and learning log-juntas (the subject of a prize offered in 2003 by Blum). As exciting as this possibility seems, it does not contradict known hardness results, nor does it upset the consensus on related problems in cryptography or complexity theory. We propose to gain more experimental and theoretical evidence for this possibility.

In practice, many classification tasks involve multiple classes. When the number of classes is large, we do not believe fast agnostic classification is possible. We posit stronger lower bounds for classification with a growing number of classes which depend on P $\neq$ NP rather than weaker conjectures about refuting random constraint satisfaction problems. We believe the problem remains challenging even when the inputs are normally distributed. This is due to close relations with the learning with errors (LWE) problem, which underpins much of modern cryptography. The difficulty of LWE and its variants depends heavily on its parameters, including the dimension of the inputs, the amount of data, the number of classes, and the amount of noise. For some choices, it is as hard as worst-case lattice problems; for others, it may be solved in polynomial time. We propose a generalization of KG for multiple classes. It may substantially improve upon logistic regression and multiclass SVM for many practical applications. It may also experimentally validate the range of safe parameters for lattice-based cryptography.

1. Introduction

This thesis studies classifiers. From initial conditions or inputs, these produce binary outcomes: either positive or negative. For example, the input may be a sample of skin tissue, or a curriculum vitae, or a financial asset; the positive outcomes may be a benign diagnosis, or acceptance of an application, or a rise in price. These outcomes may depend on observed input quantities, such as the presence of discoloration, or the number of awards conferred, or historical prices. They may also depend on unobserved quantities, in which case they are (apparently) random. Fixing an observed input x, $\mathbf{P}\left(y|x\right)$ is the conditional probability distribution of the outcome.

Perhaps the most basic hypotheses are linear. These represent inputs x as n-dimensional vectors; each component x_i is an observed numerical quantity. A linear classifier is defined by a vector w which operates upon inputs only via an inner product $\langle w, x\rangle = \sum_{i=1}^{n} w_i x_i$. The archetypal linear classifier is a halfspace, which returns an output $y \in \{-1, 1\}$ according to the sign of the inner product. That is, inputs are separated into positive and negative classes by a line passing through the origin.

$$\mathcal{H}_{\text{0-1}} = \{h_w^{\text{0-1}}(x) = \operatorname{sgn}(\langle w, x\rangle) = 2 \cdot \mathbf{1}(\langle w, x\rangle \geq 0) - 1 : w \in \mathbb{R}^n\} \tag{1}$$

The imposition of linearity is not actually a restriction, since high-dimensional linear functions may, in principle, describe essentially any phenomenon. By limiting the the number of dimensions n, linear classifiers become practically realizable, and their inaccuracies more acute.

A hypothesis should correctly predict the outcome given a new input. This is only reasonable if the new input is related to previous inputs. This motivates an assumption about how data are observed: the inputs are independently and identically generated according to a probability distribution $\mathcal{D}$. The accuracy of a classifier c is measured by its correlation, which is a rescaled negation of its error probability:

$$\chi(c) = 1 - 2 \underset{(x,y)\sim\mathcal{D}}{\mathbf{P}}\left(c(x) \neq y\right) \tag{2}$$

The correlation depends only on the conditional means of the outputs:

$$\chi(c) = \underset{x}{\mathbf{E}}\left(c(x) y_x\right) \text{ where } y_x = \mathbf{E}\left(y|x\right) \in [-1, 1]$$

Relative to a perfect correlation of 1 (corresponding to an error probability of 0), the imperfection of a set of hypotheses is:

$$\mathsf{opt} = \max_{h\in\mathcal{H}} \chi(h) \leq 1 \tag{3}$$

$\mathsf{opt} < 1$ is the purview of *agnostic* learning, a term coined by (Kearns et al., 1992) which emphasizes $\mathcal{H}$ isn't believed to be ideal. Learning versus $\mathcal{H}$ on $\mathcal{D}$ means maximizing χ: obtaining c with $\chi(c)$ close to opt. The following task, called 'agnostically learning halfspaces' or 'learning versus halfspaces', is the goal of this thesis.

$\mathcal{D}$ is a known distribution on inputs $x \in \mathbb{R}^n$ and $y_x : \mathbb{R}^n \to [-1, 1]$ are unknown outputs. The correlation of the best halfspace is:

$$\mathsf{opt} = \max_{w \in \mathbb{R}^n} \chi(h_w^{0\text{-}1}) = \max_{w \in \mathbb{R}^n} \mathop{\mathbf{E}}_{x} \left(h_w^{0\text{-}1}(x) y_x \right) < 1$$

For any multiplicative error $\alpha \in (0, 1)$ and additive error $\epsilon \in (0, 1)$, given m independent and identically distributed data $(x_1, y_1), \ldots, (x_m, y_m)$, produce a classifier c which satisfies $\chi(c) \geq (1 - \alpha)\mathsf{opt} - \epsilon$ using $\text{poly}(n, 1/\alpha, 1/\epsilon)$ data and time.

Figure 1: Agnostically learning halfspaces on a distribution $\mathcal{D}$.

1.1. Outline of this proposal

This thesis studies a new algorithm, KG, for agnostic classification. The first part introduces the algorithm in the context of previous work.

- Section 2 reviews previous methods with a critical eye; each is blocked by a substantial barrier for agnostic learning.
- Section 4 describes KG, which involves new classifiers and a new algorithm. Two theorems show KG is agnostic and uses an optimal amount of data.

The main part of the thesis studies the possibility of fast agnostic classification. This problem is intimately tied to learning parities with noise, a challenging open problem. We conjecture:

- learning versus halfspaces on normally distributed inputs (and therefore learning logarithmically-sparse parities with noise) is fast, whereas
- learning versus halfspaces on generally distributed inputs (and learning non-sparse parities with noise) is slow.

We support these conjectures with the following evidence.

- Section 3 examines the hardness of learning versus halfspaces through reductions from combinatorial optimization. Our conjectures are compatible with broad consensus in computer science theory.
- Section 4.4 experimentally evaluates KG on a variety of noisy classification problems from learning theory. It avoids barriers that previous methods do not.

The last part of the thesis concerns multiclassification with $q > 2$ classes.

- section 6.1 reviews Voronoi diagrams, the multiclass analogue of halfspaces.
- section 6.2 reviews previous algorithms for multiclassification.
- section 7 generalizes KG for multiclassification.

Multiclassification is of considerable practical interest. However, we believe fast agnostic multiclassification isn't generally possible. Section 7.1 supports this conjecture with reductions from hard problems:

- min-sum clustering reduces to learning versus Voronoi diagrams from statistical queries.
- learning with errors (with recommended values for the modulus, noise rate and noise distribution, and a potentially non-sparse secret) reduces to learning versus Voronoi diagrams.

We seek to apply KG to practically interesting problems and gain insight about the hardness of multiclassification.

As a whole, this thesis presents a novel classification algorithm of practical and theoretical interest. We propose to understand its powers and limitations, and hope to gain insight into classification itself.

2. Previous algorithms

The most basic agnostic learning algorithm picks the halfspace $h_v^{0\text{-}1}$ which maximizes correlation with the data. That is, v is the maximizer of an empirical correlation $\hat{\chi}(h_v^{0\text{-}1})$ defined by the empirical distribution $\mathcal{D}$:

$$\hat{\chi}(c) = \hat{\mathbf{E}}_{x}\left(c(x)y_x\right) = \frac{1}{m}\sum_{i=1}^{m} c(x_i)y_i \tag{4}$$

This algorithm needs a minimal amount of data, as theorem 1 (below) elaborates. However, it needs a large amount of time: the maximization is NP-hard, even to approximate (Ben-David and Simon, 2000). This difficulty motivates modifying the objective and trading data for time by choosing $c \notin \mathcal{H}$; the latter flexibility is called improper learning. This leads to four prevalent methods of learning versus linear classifiers. They are distinguished primarily by the respective sets from which c is chosen. For different real-valued functions f, they choose $c(x) = \operatorname{sgn}(f(x))$. As f becomes more complicated, the required amount of data increases; however, the optimization over f becomes simpler, so the required amount of time may decrease.

- Relaxation (such as logistic regression): let $f(x) = \langle w, x\rangle$ be a linear function, making c a halfspace. Instead of directly maximizing $\chi(\operatorname{sgn}(\langle w, \cdot\rangle))$, upper bound the sign function with a convex function, and minimize a 'relaxed' objective. Relaxation is often involved in more complicated schemes such as deep learning.
- Sampling: let $f(x)$ be a random value in $\{-1, 1\}$, equal to $\operatorname{sgn}(\langle w, x\rangle)$ with probability based on $|\langle w, x\rangle|$; this makes c a smooth halfspace. Randomly sample many w from a simple distribution in a brute-force manner, and pick the one which maximizes correlation.
- Boosting (such as AdaBoost): let f be a convex or linear combination of base classifiers: $f(x) = \sum_t a_t c_t(x)$. The resulting c is called an ensemble. Minimize a (typically) convex function of f by iteratively appending new summands to f. Picking a new

summand is called weak learning, and is left unspecified for a separate algorithm to perform.

- Lifting (such as kernel SVM and L_1 polynomial regression): let f be an element of a high-dimensional reproducing kernel Hilbert space. Choose the space so that every linear classifier h roughly corresponds to some element f. Maximizing χ over all f also maximizes χ over all h. The latter objective is non-convex, whereas the former objective is linear.

(Other classifiers, such as decision trees and nearest-neighbor rules, are popular, but are not as promising for learning versus linear classifiers.) The success of these methods depends crucially on the value of opt. If $\mathsf{opt} = 1$ — that is, a halfspace is perfectly consistent with the data — all of the above methods work correctly and use a reasonable amount of time and data. When $\mathsf{opt} < 1$, none of the aforementioned methods are satisfactory. Relaxation might not work correctly, no matter how much time and data are available; in practice, this method succumbs to noisy data or imperfect models. Sampling needs too much time, though recent techniques improve upon trivial brute-force search. Boosting needs a separate learning algorithm which must work under very general conditions; designing one is baffling in both theory and practice. Lifting works correctly, but needs an exponential amount of time and data; in practice, this method is slow or overfits.

This section critically reviews each of these methods. A growing body of evidence suggests they are not promising for learning versus halfspaces. Nonetheless, each method contributes a critical component to our new algorithm:

- section 2.1: averaging the data to produce a useful halfspace,
- section 2.2: smoothing halfspaces to obtain a differentiable function,
- section 2.3: gradually reweighting the data to produce a new correlation,
- section 2.4: restricting the distribution of the inputs.

2.1. Halfspaces via relaxation

Halfspaces are convenient because they are easy to represent (as n numbers) and do not need much data to train. The following result is a consequence of the fundamental theorem of statistical learning due to Vapnik and Chervonenkis.

Theorem 1 *Let $\epsilon \in (0,1)$ and $m \geq \tilde{O}(n/\epsilon^2)$. For any distribution $\mathcal{D}$ on $\mathbb{R}^n$ and any outputs $y_x : \mathbb{R}^n \to [-1,1]$:*

- *the empirical and true correlations of any halfspace are close: $\sup_{w\in\mathbb{R}^n} |\chi(h_w) - \hat{\chi}(h_w)| \leq \epsilon$.*
- *learning is achievable by empirical correlation maximization: if w maximizes $\hat{\chi}(h_w)$, then $\chi(h_w) \geq \mathsf{opt} - \epsilon$.*

(Shalev-Shwartz and Ben-David (2014) theorems 6.8 and 9.2).

To avoid the nonconvexity of eq. (4), the most popular classification algorithms instead minimize a convex upper bound on $-\chi$. With binary outputs, each term $-\text{sgn}(\langle w, x\rangle)y_x = \text{sgn}(-\langle w, x\rangle\, y_x)$ may be written in terms of the margin $\langle w, x\rangle\, y_x$, the amount by which w is correct. Upper bound each nonconvex $\text{sgn}(\cdot)$ with a convex 'loss' function $\ell : \mathbb{R} \to \mathbb{R}$. Using a piecewise-linear 'hinge' function leads to support vector machines, perceptrons, and winnow (Zhang, 2001); the cross-entropy function leads to logistic regression; Mahalanobis distances lead to linear discriminant analysis (Park and Park, 2005). A linear approximation (though not an upper bound) leads to a simple joint average of the data:

$$\mathop{\mathbf{E}}_{x}\,(x \cdot y_x)$$

This is the most elemental vector in relaxation. If it is zero, then all relaxations fail in the sense that the minimizer of the convex loss is always zero. Two applications of convexity contradict the existence of a solution w with lower loss:

$$0 < \ell(0) - \mathop{\mathbf{E}}_{x}\,(\ell(\langle w, x\rangle\, y_x)) \le \ell(0) - \ell(\mathop{\mathbf{E}}_{x}\,(\langle w, x\rangle\, y_x)) \le -\ell'(0)\mathop{\mathbf{E}}_{x}\,(\langle w, x\rangle\, y_x) = -\ell'(0)0 \quad (5)$$

(The first inequality is the assumption to be contradicted; the second is the zeroth order definition of convexity; the third is the first-order definition of convexity. Substituting the zero joint average yields the contradiction.) When the inputs are standard normals, the average coincides with the Fisher linear discriminant (Fisher, 1936), perhaps the first linear classifier ever used. It was rediscovered as the 'averaging algorithm' (Servedio and Valiant, 2001) and tolerates some noise in the outputs when the inputs are distributed uniformly on the sphere. Among convex relaxations, it is the most robust to certain kinds of noise (van Rooyen and Krishna Menon, 2015).

Since the margin $f(x)y_x$ is signed, it is sensitive to misclassification: if $|f(x)|$ is large, flipping the sign of y_x affects the relaxation substantially more than the correlation. Unfortunately, this makes relaxation susceptible to to noisy outliers. For any convex ℓ, the resulting $h_w^{0\text{-}1}$ may be poor, even if a halfspace is almost consistent with the data. There are distributions for which $\chi(h_w^{0\text{-}1}) \le \epsilon$ even though $\mathsf{opt} \ge 1-\epsilon$, for arbitrarily small ϵ (Ben-david et al., 2012). Similar results hold even when the inputs are distributed uniformly on the circle (Awasthi et al. (2015a) theorem 6) or are far from the separating hyperplane (Long and Servedio, 2010, 2011). The joint average may have poor correlation even if the inputs are normally distributed.

Theorem 2 *Let $\mathcal{D}$ be the normal distribution. There are outputs y_x for which, despite being consistent with a halfspace except for some probability $(1-\beta)/2 = \eta > 0$, the joint average is suboptimal: $\chi(h_w^{0\text{-}1}) \le \mathsf{opt} - \Omega(\beta(1-\beta)/(1+\beta))$ (Awasthi et al. (2015a) theorem 5.)*

These results, along with experiments in section 4.4, show small amounts of coordinated noise can pull w away from the optimum.

Mitigating noisy outliers has been extensively studied.

- (Kalai et al., 2008) eliminates inputs that are too close to one another,
- (Klivans et al., 2009) eliminates inputs directions of 'suspiciously' high variance,

- (Awasthi et al., 2014) reweights inputs rather than removing them, and doesn't bother with inputs that are close to boundary.

These algorithms fortify relaxation against noise. However, as section 3 elaborates, they tolerate just some η fraction of the outputs to be inconsistent with a halfspace, and η cannot be arbitrarily close to 1/2.

2.2. Smooth halfspaces via sampling

Maximizing eq. (4) with random, brute-force search takes $n^{O(1/\epsilon)}$ time. The recent technique of (Zhang et al., 2015) cleverly eliminates the exponential dependence on the dimension n. Like relaxation, it approximates eq. (4), but obtains smoothness (in the sense of Lipschitz continuity) rather than convexity. It replaces the sign function $\mathrm{sgn}(a)$ in halfspaces with a smooth 'sigmoid' function ψ of slope at most L. A typical choice is the logistic sigmoid function $\frac{1}{1+e^{-4La}}$. The resulting smooth halfspaces take values in $[-1, 1]$:

$$\mathcal{H} = \{h_w(x) = \psi(\langle w, x\rangle) = |h_w(x)| \cdot h_w^{0\text{-}1}(x) : w \in \mathbb{R}^n\} \tag{6}$$

The magnitude of the output can be used as a probability for a randomized classifier which operates as follows:

> With probability $|h_w(x)|$, return $\mathrm{sgn}(\langle w, x\rangle)$. Otherwise guess -1 or 1 uniformly at random.

The correlation of h_w equals the correlation of this randomized classifier. The correlation of a smooth halfspace is closely related to the margin correlation of a halfspace, which is the probability an input is correctly classified and farther than γ from the boundary:

$$\chi_\gamma(w) = \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} (\mathbf{1}(\langle w, x\rangle \geq 0) \neq y \ \wedge\ |\langle w, x\rangle| \geq \gamma) \tag{7}$$

The margin parameter γ is essentially the inverse of the slope L. Their quantitative relationship is examined in (Shalev-Shwartz et al., 2011; Ben-David and Simon, 2000; Birnbaum and Shalev-Shwartz, 2012).

Smoothness makes the required amount of data independent of the dimension. This is quantified by Rademacher complexity: the expected maximum correlation, taken over a set of functions $\mathcal{F}$, with m uniformly random outputs $\sigma_1, \ldots, \sigma_m$ on m inputs $x_1, \ldots, x_m$ drawn from $\mathcal{D}$.

$$\mathcal{R}(\mathcal{F}) = \mathop{\mathbf{E}}_{\substack{x_1,\ldots,x_m\sim\mathcal{D}\\ \sigma_1,\ldots,\sigma_m\sim\{-1,1\}}} \left(\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} f(x_i)\sigma_i\right) := \mathop{\mathbf{E}}_{\hat{\mathcal{D}},\sigma} \left(\sup_{f\in\mathcal{F}} \mathop{\hat{\mathbf{E}}}_{x} (f(x)\sigma_x)\right) \tag{8}$$

Bounding the Rademacher complexity bounds the required amount of data.

Theorem 3 *With probability $1-\delta$ over the sample $\hat{D} \sim \mathcal{D}$:*

$$m \geq \frac{8\log(1/\delta)}{(4\mathcal{R}(\mathcal{F}) - \varepsilon)^2} \implies \sup_{f\in\mathcal{F}} |\chi(f) - \hat{\chi}(f)| \leq \varepsilon$$

(Boucheron et al. (2005) Theorem 3.2).

```
For $t = 1, \ldots, T$:
  Randomly subsample $\tilde{m} < m$ inputs
  Uniformly sample $\tilde{y}$ from $\{-1,1\}^{\tilde{m}}$
  Set $w_t = \underset{||w|| \leq 1}{\text{argmin}} \underset{x,y}{\tilde{\mathbf{E}}} \left( (\langle w, x \rangle - \tilde{y})^2 \right)$
Return $w = \underset{1 \leq t \leq T}{\text{argmax}} \, \hat{\chi}(h_{w_t})$
```

Figure 2: The least-squares initialization algorithm of (Zhang et al., 2015).

The Rademacher complexity may be bounded by the slope, independently of the dimension and distribution of the inputs.

Theorem 4 *For any $L > 0$ and inputs $||x|| \leq \beta$,*

$$\mathcal{R}_L = \sup_{\mathcal{D}} \mathcal{R}(\mathcal{H}) \leq L\beta \sqrt{\frac{2}{m}}$$

(Kakade et al. (2008) Theorem 1 and Example 3.1.1).

This theorem ensures a small ($\tilde{m} < m$) subsample of data are representative of all the data. If $\tilde{m}$ is small enough, it is plausible to randomly guess outputs $\tilde{y}$ that match those of $h_v^{\text{0-1}}$ on the subsample; if this happens, $||w_t - v||$ is small, making $\hat{\chi}(h_v) - \hat{\chi}(h_{w_t})$ small. This is the basis of the least-squares initialization algorithm of (Zhang et al., 2015).

Theorem 5 *Let $\mathcal{D}$ be any distribution on inputs $||x|| \leq 1$. Define opt relative to smooth halfspaces with vectors $||w|| \leq 1$ and a sigmoid ψ of slope L. The least-squares initialization algorithm returns w satisfying $\chi(h_w) \geq \mathsf{opt} - 22L \cdot \epsilon$ using $m = O(1/\epsilon^2)$ data and $\text{poly}(m, n, e^{(2/\epsilon^2)\log(1/\epsilon)}, \log(1/\delta))$ time (Zhang et al. (2015) theorem 2).*

Unlike relaxation, the smooth approximation of eq. (4) is reliable; the algorithm learns versus halfspaces as $L \to \infty$. However, even for fixed L, no algorithm finds a smooth halfspace with high correlation in polynomial time.

Theorem 6 *Let ψ be a piecewise-linear function with $L = 1$, and define opt relative to smooth halfspaces with vectors $||w|| \leq 1$. If $\text{RP} \neq \text{NP}$, finding w satisfying $\chi(h_w) \geq \mathsf{opt} - \epsilon$ in randomized $\text{poly}(n, 1/\epsilon)$ time is impossible. (Zhang et al. (2015) proposition 1)*

2.3. Ensembles via boosting

A natural way to expand the set of hypotheses, and thereby trade between data and time, is by combining multiple hypotheses. Smooth halfspaces, as defined in section 2.2, are particularly amenable to combination. The Rademacher complexity of the convex hull of $\mathcal{H}$ equals that of $\mathcal{H}$, so theorem 4 holds for averages of smooth halfspaces (Bartlett and Mendelson, 2003). For general hypotheses (with potentially unbounded Rademacher complexity), the additional amount of data required for combinations depends upon the algorithm producing the combinations (Telgarsky, 2012).

Boosting is a high-level approach for (linearly) combining hypotheses as $f = \sum_t a_t h_t$. It is not the basis of any state-of-the-art algorithms for agnostically learning halfspaces. Instead of reviewing quantitative guarantees, this section highlights some qualitative aspects of boosting which differ from our new algorithm. As in relaxation or sampling, boosting approximates the objective eq. (4) with a 'potential' function. f is iteratively constructed by:

- finding a hypothesis which decreases the potential ('weak learning'), and
- increasing the probability of data which are misclassified ('reweighting').

Nonconvex potentials retain a substantial computational burden; for example, the reweightings in BrownBoost (Freund, 2001) and ArchBoost (Hanbo Li and Bradic, 2015) can be hard to compute. Convex potentials are vulnerable to the hazards described in section 2.1; they are susceptible to noisy outliers since they lead to reweightings based on the margin of misclassification (Long and Servedio, 2010). This includes multiplicative weights, the archetypal reweighting scheme:

$$y_x \to e^{-\beta \langle w_t, x \rangle y_x} y_x \tag{9}$$

for a small step size $\beta > 0$.

A joint distribution over inputs and outputs is weakly learnable if there is a nontrivial hypothesis: $\max_{h \in \mathcal{H}} \chi_t(h) > 0$. The weak learning assumption is that for some edge $\gamma > 0$, all reweighted distributions are weakly learnable. Weak learning algorithms are challenging to design. This difficulty has a mathematical explanation if the potential function is convex, since boosting is interpretable as a primal-dual algorithm for minimizing the potential over all f. The dual attempts to find a distribution which is uncorrelated with all h. All dual-feasible distributions are not weakly learnable; that is, the weak learning assumption (i.e. when boosting 'works') implies a degeneracy of the dual problem (Telgarsky, 2012). In learning terminology, boosting relies on linear separability (Shalev-Shwartz and Singer, 2010).

Gradually reweighting the inputs can fortify boosting against noise (Domingo and Watanabe, 2000; Servedio, 2003). This approach underlies a variety of agnostic boosting algorithms (Ben-David et al., 2001; Kanade and Kalai, 2009; Chen et al., 2015). The empirical correlation $\hat{\chi}$ weights each input equally. A reweighting is smooth if it does not put substantial weight on any single input. Smooth boosting algorithms involve only smooth correlations; the dual optimization typically involves projection onto the set of smooth distributions (Barak et al., 2009). Weak learnability of smooth distributions is equivalent to separability of nearly all of the data (Shalev-Shwartz and Singer, 2010). Unfortunately, if the amount of noise is large, smooth boosting algorithms are slow; section 3 includes quantitative assessments.

2.4. Polynomials via lifting

Lifting is currently the only general, systematic way to design agnostic learning algorithms. It reduces algorithm design to function approximation, and underlies the fastest algorithms for learning versus intersections of halfspaces (Harsha et al., 2013; Kane et al., 2013), disjunctions (Feldman and Kothari, 2014), submodular functions (Cheraghchi et al., 2012), polynomial threshold functions (Harsha et al., 2014; Diakonikolas et al., 2010), and convex sets (Klivans et al., 2008). For learning versus (smooth) halfspaces in n dimensions up to a $(1-\alpha)$-approximation of opt, it uses $\text{poly}(m, n, 1/\epsilon)$ time for different, super-polynomial amounts m of data.

- Versus halfspaces on uniformly distributed inputs, with $\alpha = 0$, polynomial regression uses $m = \text{poly}(n^{1/\epsilon^2})$ data (Kalai et al., 2008).
- Versus smooth halfspaces on generally distributed inputs, for $10 \leq \alpha \leq L$, $\tau = L/\alpha$, and some constant C, kernel SVM uses roughly $m = O(\frac{\exp(C\tau \min(\tau, \log L))}{\epsilon^2})$ data (Birnbaum and Shalev-Shwartz, 2012).
- Versus halfspaces on uniformly distributed inputs, polynomial regression combined with 'localization' uses $m = \text{poly}\left(n^{\frac{\log^3(1/\alpha)}{\alpha^2}}, \log(\frac{1}{\epsilon})\right)$ data (Daniely, 2014).

Lifting is a venerable idea; the same basic algorithm has been used since the 1960s, and much of it was proposed by Gauss and Legendre in the early 1800s. The main step is fitting a degree-d polynomial f (i.e., an element of $\mathcal{P}_d$) to the data:

$$\min_{f \in \mathcal{P}_d} \ell(f) = \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} \left((f(x) - y)^2\right) \tag{10}$$

It is easy to bound the amount of time this takes. Equation (10) is a convex optimization problem, whose size is exponential in d, when f is represented as a vector of coefficients, one for each monomial. It can be solved in polynomial time if d is a constant.

By contrast, the justification for replacing the hypotheses $\mathcal{H}$ with $\mathcal{P}_d$ is a challenging, ongoing research topic. It is addressed by various notions of polynomial approximation. The threshold degree of h is the minimal degree d required to express $h(x) = \text{sgn}(f(x))$. If it is constant for all $h \in \mathcal{H}$, and $\mathsf{opt} = 0$, then polynomial regression ultimately finds a correct f. In this scenario, eq. (10) is solved by Rosenblatt's 1957 perceptron, arguably the first learning algorithm (Rosenblatt, 1957). Minsky and Papert initiated the study of threshold degree (Minsky and Papert, 1972). Their lower bounds, especially for parity functions, were largely responsible for an artificial intelligence 'winter' from 1974—1980 (Russell and Norvig, 2003).

In 1993, Linial, Mansour and Nisan proposed a more forgiving notion of approximation in terms of the distribution of the inputs (Linial et al., 1993):

$$\forall h \in \mathcal{H}, \exists f \in \mathcal{P}_d \quad \text{s.t.} \quad \mathop{\mathbf{E}}_{x\sim\mathcal{D}_x} \left((h(x) - f(x))^2\right) \leq \alpha^2 \tag{11}$$

They showed depth-Δ, size-s boolean circuits satisfy eq. (11) (for polynomially large α) when D_x is the uniform distribution on the boolean hypercube. They utilize a connection between eq. (11) and the Fourier-analytic properties of $\mathcal{H}$ on the uniform distribution. This approach has been successfully extended to product distributions (Blais et al., 2010), permutation-invariant distributions (Wimmer, 2010), and Markov random fields (Kanade and Mossel, 2015). In 2005, Kalai, Klivans, Mansour, and Servedio slightly modified the polynomial regression algorithm for agnostic learning (Kalai et al., 2008). They replace the L_2 distance in eq. (10) with the L_1 distance.

1. Fit a degree-d polynomial p to the data:

$$\min_{p \in \mathcal{P}_d} \ell(p) = \mathbf{E}\left(|p(x) - y_x|\right) \tag{12}$$

2. If the outputs are binary, with the convention that -1 and 1 are true and false, use p as a threshold in $c(x) = \operatorname{sgn}(p(x) + t)$. Choose t to minimize $\ell(c)$.

They demand a similarly modified approximation:

$$\forall h \in \mathcal{H}, \exists p \in \mathcal{P}_d \quad \text{s.t.} \quad \mathop{\mathbf{E}}_{x \sim \mathcal{D}}\left(|p(x) - y_x|\right) \leq \alpha \tag{13}$$

Equation (11) implies eq. (13); recently, (Kane et al., 2013) showed how to directly prove eq. (13) when $\mathcal{D}$ is log-concave. A simple argument shows $\ell(c) \leq \frac{1}{2}\ell(f)$, so minimization of eq. (12) is appropriate. The triangle inequality yields $\ell(f) \leq \mathop{\mathbf{E}}_{(x,y) \sim \mathcal{D}}(|y - h(x)| + |h(x) - f(x)|)$. Combining the previous inequalities and picking $h \in \mathcal{H}$ to achieve $\ell(h) = \mathsf{opt}$ yields $\ell(c) \leq \mathsf{opt} + \alpha/2$.

In 2010, Shalev-Shwartz, Shamir, and Sridharan replaced $\mathcal{P}_d$ with infinite degree polynomials of bounded norm (Shalev-Shwartz et al., 2011). Fitting such a polynomial is tractable. Rather than minimizing ℓ directly, draw m data and formulate an 'empirical' objective:

$$\min_{f \in \mathcal{P}_B} \hat{\ell}(f) = \sum_{i=1}^{m} |f(x_i) - y_i| \tag{14}$$

$\hat{\ell} \to \ell$ as m increases. $\mathcal{P}_B$ is a ball in a Hilbert space spanned by elements $\{\phi(x) : x \in \mathbb{R}^n\}$ whose inner products can be easily computed: $\langle \phi(x), \phi(x') \rangle = \frac{1}{1 - \frac{1}{2}\langle x, x' \rangle}$. An important consequence is that the solution of eq. (14) can be written as $f(x_i) = \sum_{j=1}^{m} c_j \langle \phi(x_i), \phi(x_j) \rangle$. By precomputing the inner products $\langle \phi(x_i), \phi(x_j) \rangle$ and optimizing over the coefficients c_j, eq. (14) becomes a convex optimization problem of size m. With the power of infinite-degree polynomials, (Shalev-Shwartz et al., 2011) obtains a non-probabilistic approximation akin to those studied before (Linial et al., 1993). Their analysis is limited to smooth halfspaces (recall eq. (6)) with sigmoids ψ, unlike (Linial et al., 1993). ψ is called an activation or transfer function. Approximating h_w reduces to approximating the univariate function ψ, which is amenable to Chebyshev interpolation. In particular, the univariate polynomials

$$\mathcal{U}_B = \left\{ u : \sum_{d=0}^{\infty} \sum_{|\alpha|=d} 2^d u_\alpha^2 \leq B \right\}$$

define multivariate polynomials contained in $\mathcal{P}_B$:

$$\{f(x) = u(\langle w, x \rangle) : w \in \mathbb{R}^n, u \in U_B\} \subseteq \mathcal{P}_B$$

The univariate polynomials approximate various ψ in the following sense:

$$\forall x, w \in \mathbb{R}^n, \ \exists u \in \mathcal{U}_B \quad \text{s.t.} \quad |\psi(\langle w, x \rangle) - u(\langle w, x \rangle)| \leq \alpha$$

This univariate-linear analysis is less general than the prior analysis of polynomial regression.

In practice, L_1 polynomial regression, with either $\mathcal{P}_d$ or $\mathcal{P}_B$, uses too much time and data. When using $\mathcal{P}_d$, the size of eq. (12) and the required amount of data are both exponential in d; even $d = 2$ is usually impractical. When using $\mathcal{P}_B$, m is exponential in the Lipschitz constant of ψ. Efficiently solving problems like eq. (14) is a long-standing goal of 'kernel methods' in machine learning. Many interesting approaches have been developed (Le et al., 2013; Cotter et al., 2012). However, to the best of our knowledge, no practical method approximately solves eq. (14) when m is very large.

Limits on the approximation power of $\mathcal{P}_d$ are well known. Superconstant degree lower bounds for parities, linear-size boolean formulae, and ANDs of majorities have been known since the 1960s (Minsky and Papert, 1972). Constant-degree polynomials cannot approximate boolean functions where any single input bit has too much 'influence' on the output (Ben-Eliezer et al., 2009). The crucial question is whether the limits of polynomials coincide with the limits of learning. This is essentially the case when the inputs have independent (but not necessarily identical) boolean coordinates, and the hypotheses have non-trivial dimension. Say a function is d-resilient if it is uncorrelated (in the usual sense of eq. (2)) with any degree-d parity function. If all hypotheses are far from d-resilient, then they may be approximated by degree-d polynomials; if there is a d-resilient hypothesis then agnostic learning requires $n^{\Omega(d)}$ statistical queries (Dachman-Soled et al., 2015). This rough equivalence of polynomial approximation and agnostic learning does not extend to other distributions; polynomial regression is known to fail when other algorithms succeed. For inputs with heavier tails than the (singly-exponential) Laplace distribution, there is an ϵ such that no polynomial (of any degree) can approximate the sign function (Bun and Steinke, 2015). When $n = 1$, empirical risk minimization is efficient (by simple brute-force search) and successful, but polynomial regression is not.

Lifting generalizes beyond polynomial f to RKHS elements. This generality is beneficial: disjunctions on permutation-invariant distributions may be approximated by a reasonably-sized basis of functions, but not by low-degree polynomials (Feldman and Kothari, 2014). However, even the most general choices of f cannot be effective. The following lower bounds are phrased in terms of the margin γ (recall eq. (7)).

- Choosing f to be a vector of moderate dimension which minimizes some convex function subject to convex constraints. If $\alpha = 0$, then this approach must take $\exp(1/\gamma)$ time in the worst case. It runs in polynomial time only if $\alpha = \Omega(\frac{\sqrt{1/\gamma}}{\text{poly}(\log 1/\gamma)})$ (Daniely et al., 2012).

- Choosing f to be an element of a reproducing kernel Hilbert space which minimizes some convex function subject to a norm constraint. If $\alpha = 0$, then this approach must take $\exp(1/\gamma)$ time in the worst case. It runs in polynomial time only if $\alpha = \Omega(\frac{1/\gamma}{\text{poly}(\log 1/\gamma)})$ (Daniely et al., 2012).

Rather than lifting the nonconvex optimization over w to a high-dimensional convex optimization, it is possible to lift to a moderate dimension optimization which, though nonconvex, may be easier to solve. This approach has been proposed as a theoretical model of deep learning:

1. Approximate eq. (4) as a deep network f_W with parameters $W \in \mathbb{R}^N$.

2. Find a local optimum of $\chi(f_W)$, preferably with a method which avoids saddle points (Dauphin et al., 2014).

If N is sufficiently large, then most local optima of $\chi(f_W)$ seem to have correlation close to $\chi(f_{W^*})$ in practice (Dauphin et al., 2014). Under strong assumptions on the relationships of the parameters, (Choromanska et al., 2014) proves an asymptotic tradeoff between N and the quality of the local optima. It is not clear if these results can be formally strengthened for agnostic learning. In practice, deep networks seem susceptible to limited forms of noise (Goodfellow et al., 2014).

3. Learning versus halfspaces

This section reviews the difficulty of learning versus halfspaces. This problem is meaningful only if $\mathsf{opt} > 0$ — that is, the binary outputs are consistent with a halfspace with probability $1 - \eta$, where $\eta \in (0, 1/2)$ is a 'noise' or 'inconsistency' rate. If strong assumptions are made about the manner of inconsistency, then learning is easy. In the following scenarios, the inconsistent outputs are either completely random or structured in a known way.

- random classification noise flips the sign of each output with probability $\eta \in (0, 1/2)$ independently of the input. This is the subject of the first model for noise-tolerant learning (Angluin and Laird, 1988), as well as the first noise-tolerant learning algorithm for halfspaces (Blum et al., 1998).

- if a halfspace has margin correlation γ (recall eq. (7)), then relaxation (with hinge loss) obtains a halfspace with margin correlation at least $\gamma/2$ (Ben-david et al., 2012).

- monotonic noise requires inconsistency to diminish with the margin; that is, noisy data must be near the decision boundary (Bylander, 1998). The average algorithm copes with monotonic noise (Servedio and Valiant, 2001).

- single link outputs, wherein $y_x = u(\langle v, x \rangle)$ for some v and nondecreasing u, are a generalization of monotonic noise. Isotonic regression copes with this scenario (Kalai and Sastry, 2009).

As assumptions about the inconsistent outputs are lifted, learning becomes more challenging. In the following scenarios, the inconsistency has no discernible structure. A fast algorithm tolerates the noise rate η if it returns w satisfying $\chi(h_w) \geq \mathsf{opt} - \epsilon$ in time poly $\left(n, \frac{1}{\epsilon}, \frac{1}{1-2/eta}\right)$.

- bounded (Massart) noise flips the sign of y_x with probability $\eta(x) \in (0, 1/2)$, which potentially depends on the input. For inputs distributed uniformly on the sphere, (Awasthi et al., 2015a) tolerates any $\eta \in (0, 1/2)$.

- adversarial noise: with probability η, data is sampled from an arbitrary (but fixed) distribution. (Awasthi et al., 2014) tolerates $\eta = \Omega(\epsilon/\log^2(1/\epsilon))$.

- malicious noise: with probability $\eta \in (0, 1/2)$, both inputs and outputs are provided by an adaptive adversary. (Since this changes the input distribution, it is more general than agnostic learning.) (Awasthi et al., 2014) tolerates $\eta = \Omega(\epsilon/\log^2(1/\epsilon))$.

Agnostic learning eschews assumptions about the outputs. If no assumptions are made about the input distribution $\mathcal{D}$, then learning versus halfspaces is as hard as solving fundamental lattice problems.

Theorem 7 *Let $\mathcal{D}$ be arbitrary and $\alpha = 0$. Assume the shortest vector problem with parameter $\tilde{O}(n^{1.5})$ cannot be solved in polynomial time. Learning versus $\mathcal{H}_{0\text{-}1}$ in $\mathrm{poly}(n, 1/\epsilon)$ time is impossible. Also, learning versus $\mathcal{H}$ in $\mathrm{poly}(L, 1/\epsilon)$ time is impossible (Shalev-Shwartz et al., 2011).*

Approximate learning is as hard as refuting random constraint satisfaction problems.

Theorem 8 *Let $\mathcal{D}$ be uniform on $\{-1, 1\}^n$ and $\alpha \in (0, 1)$ be constant. Under the random k-XOR assumption, learning versus $\mathcal{H}_{0\text{-}1}$ in $\mathrm{poly}(n, 1/\epsilon)$ time is impossible. (Daniely, 2015)*

This theorem avoids the strong random CSP assumption, as used in (Daniely et al., 2013) and subsequently invalidated in (Allen et al., 2015).

4. Our contributions

This section introduces our new approach to classification, which consists of new classifiers and a new learning algorithm.

4.1. New classifiers

Recall the construction of smooth halfspaces in section 2.2, which approximate the sign function $\mathrm{sgn}(a)$ with a sigmoid function of slope at most L. Instead of the usual 'logistic' sigmoid, KG uses a sigmoid derived from the Laplace distribution:

$$\psi(a) = \begin{cases} 1 - e^{-La} & a \geq 0 \\ -1 + e^{La}, & \text{otherwise} \end{cases}$$

This function is numerically stable and twice differentiable.

$$\begin{aligned} |\psi(a)| &= 1 - e^{-L|a|} \\ \psi'(a) &= Le^{-L|a|} = L(1 - |\psi(a)|) \\ \psi''(a) &= -\mathrm{sgn}(a)L^2 e^{-L|a|} \end{aligned}$$

Despite its simplicity and numerical appeal, the Laplace sigmoid is rare in machine learning literature. We note that smooth approximations of the sign function are the subject of continued interest (Kamrul Hasan and Pal, 2015).

Smooth lists of halfspaces are randomized classifiers that naturally generalize smooth halfspaces. They are defined by lists of vectors $w_1, \ldots, w_T$. Given an input x, they operate as follows:

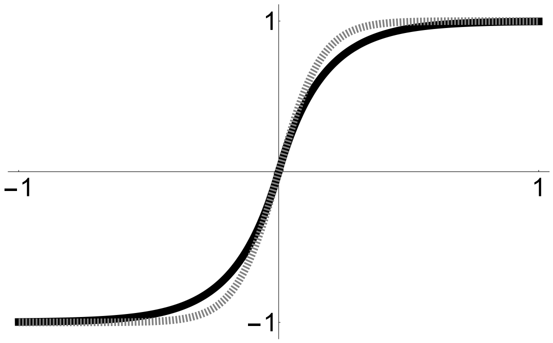

Figure 3: With $L = 6$, the Laplace sigmoid (black) and the logistic sigmoid (dashed).

```
For $t = 1, \ldots, T$:
  With probability $|h_{w_t}(x)| = 1 - e^{-L|\langle w_t, x\rangle|}$, return $\operatorname{sgn}(\langle w_t, x\rangle)$.
Guess $-1$ or $1$ uniformly at random.
```

This randomized classifier has the same correlation as the following real-valued function $f_{w_1,\ldots,w_T}(x)$, defined recursively:

$$f_\emptyset(x) = 0 \qquad f_{w_1,\ldots,w_T}(x) = h_{w_1}(x) + (1 - |h_{w_1}(x)|) f_{w_2,\ldots,w_T}(x) \tag{15}$$

Here is how smooth lists relate to other classifiers.

- Smooth lists are similar to decision lists (Rivest, 1987), which operate as follows:

  ```
  For $t = 1, \ldots, T$:
    If the deterministic function $\pi_t(x) = 1$, return the fixed value $v_t \in \{-1, 1\}$.
  Return 1.
  ```

  Decision lists are more expressive than smooth lists. For example, if π_t are halfspaces, decision lists are intersections of halfspaces, which cannot be represented as smooth lists. However, decision lists are not convenient to train. Their complexity grows with their length, which discourages appending many elements. (This difficulty was overcome by the notable algorithm of (Blum et al., 1998), which is described in section 4.2.) As the next section proves, the complexity of a smooth list is independent of its length. When $\mathsf{opt} = 1$, previous algorithms learn $\mathcal{D}$ versus decision lists, possibly returning different kinds of classifiers (Klivans and Servedio, 2006).

- If each $v_t = -1$, then a decision list is called a cascade of classifiers (Viola and Jones, 2001). These can be fast to evaluate in applications with imbalanced outputs, such as computer vision: obvious inputs are classified early, and further processing is reserved for the occasional output 1. Smooth lists behave similarly, even without imbalanced outputs: inputs far from decision boundaries tend to be classified earlier. However, we focus on the time needed to train smooth lists, not evaluate them.

- Feedforward neural networks also involve a sequence of sigmoid functions. Neural networks typically transform the input x to another vector x'. Smooth lists are more comparable to the final classification layer of a neural network, which is typically a linear classifier learned by relaxation.

- In general, hypotheses are often averaged according to a probability distribution. Such combination are ensembles, as described in section 2.3. Smooth lists are not ensembles, since the distribution of which classifier returns depends on the input x.

4.1.1. Decomposition and data

This section shows that an amount of data large enough to agnostically learn linear classifiers also suffices for smooth lists of halfspaces. Let $\mathcal{F}$ be the set of norm-bounded smooth lists:

$$\mathcal{F} = \{f_{w_1,\ldots} : \sum_t ||w_t|| \leq L\}$$

The proof of the following result is in the appendix. It exploits the distribution independence of the bounds in section 2.2.

Theorem 9 *Recall the Rademacher complexity $\mathcal{R}_L$ of smooth halfspaces in theorem 4. For all distributions $\mathcal{D}$,*

$$\mathcal{R}_L = \mathcal{R}(\mathcal{F})$$

Therefore, under the conditions of theorem 11,

$$\sup_{f \in \mathcal{F}} (\chi(f) - \hat{\chi}(f)) \leq \epsilon$$

with probability $1 - \delta$ over the sample $\hat{D} \sim \mathcal{D}$.

The preceding result is tight (with $T = 1$) but pessimistic. Intuitively, a smooth halfspace at the end of a list has lower complexity than an independent one, but the bound does not reflect this. Algebraically, later correlations have decreasing weight (in the lower-magnitude outputs), but the proof does not exploit this. Further assumptions on $\mathcal{D}$ may control the decreasing weights and lead to a tighter bound.

Smooth lists of halfspaces are useful because of their relation to smooth halfspaces. Clearly a smooth halfspace is a smooth list of length 1. Interestingly, the smooth halfspace defined by w can be decomposed into an arbitrary-length smooth list of halfspaces defined by scalings of w.

Theorem 10 *$h_w = f_{\beta_1 w,\ldots,\beta_T w}$ for any $\beta_1, \ldots, \beta_T$ satisfying $\beta_t \geq 0$ and $\sum_{t=1}^{T} \beta_t = 1$.*

This decomposition enables an iterative algorithm and competitive analysis.

4.2. New learning algorithm

The KG algorithm (fig. 4) trains smooth lists the same way they are used. It appends a vector to the list, reweights the data by the probability they would pass to the next vector, and repeats. Parts of this algorithm are reminiscent of previous ones designed to resist noise.

```
For $t = 1, \ldots, T$:
    $w_t = \frac{1}{m} \sum_{i=1}^{m} x_i y_i$
    $w_t = \beta_t (w_t / ||w_t||)$
    $y_i = e^{-|\langle w_t, x_i \rangle|} y_i$
Return $w_1, \ldots, w_T$
```

```
For $t = 1, \ldots, T$:
    With probability $1 - e^{-|\langle w_t, x \rangle|}$, return $\mathrm{sgn}(\langle w_t, x \rangle)$
Return $-1$ or $1$ uniformly at random.
```

Figure 4: The training algorithm (left) operates upon data $\{x_i, y_i\}_{i=1}^{m}$ for T iterations. It involves a sequence of positive scales $\{\beta_t\}_{t=1}^{T}$. On each iteration, it computes, rescales, and stores the average of all the data. It reduces the weight of data which are similar to this average. The weight is interpreted as a passing probability in the classification algorithm (right), which operates upon an input x. A stored average is used to classify an input if they are similar; otherwise, the input is passed to the next average.

- (Klivans et al., 2009) computes the same average vector w_t at each iteration. However, it forms a combination of halfspaces via boosting rather than a smooth list. The reweighting differs accordingly.

- (Blum et al., 1998) produces a decision list (as defined in section 4.1) of halfspaces defined by vectors $w_1, \ldots, w_T$. It returns $\mathrm{sgn}(\langle w_t, x \rangle)$ if $|\langle w_t, x \rangle|$ is larger than some threshold, and otherwise proceeds to the next element. It trains each halfspace with the perceptron algorithm upon a subset of the data which would have (on average) large margin; it passes the remaining data to subsequent steps. KG smooths the 'return' event and picks a vector according to a simpler, more conservative criterion.

Unlike its practical forbears, relaxation and boosting, we have strong evidence that KG is a correct agnostic learning algorithm. Unlike lifting, it uses an optimal amount of data. We have a nearly complete proof of the following theorem.

Conjecture 11 *Let $\mathcal{D}$ be any distribution on bounded inputs $||x|| \leq 1$ and outputs $|y_x| \leq 1$. Define opt relative to smooth halfspaces, per eq. (6), with slope L and vectors $||w|| \leq 1$. For any $\epsilon \in (0, 1)$, the algorithm uses $m = O(L^2/\epsilon^2)$ data and $O(m^2(T + k))$ time, assuming inner products $\langle x_i, x_j \rangle$ between inputs can be computed in $O(k)$ time. For some positive sequence $\{\beta_t\}_{t=1}^{T}$, the resulting classifier c satisfies $\chi(c) \geq \mathsf{opt} - \epsilon$ for sufficiently large T.*

Rather than explicitly computing the averages w_t by manipulating the inputs, it is possible to directly solve for the inner products used to classify and reweight data:

$$y_x \leftarrow e^{-\beta|\langle g, x \rangle|} y_x$$

This dual update can be expressed in terms of the kernel matrix $K_{x,x'} = \langle x, x' \rangle$, the norm of the joint sum $G = ||\sum_x x \cdot y_x||$, and the unit vector g in that direction.

$$\begin{aligned}
(Ky)_x &= \sum_{x'} K_{x,x'} y_{x'} \\
G^2 &= \left\langle \sum_x y_x x, \sum_{x'} y_{x'} x' \right\rangle = \sum_{x,x'} y_x y_{x'} K_{x,x'} = y^T K y = ||y||_K^2 \\
\langle Gg, x \rangle &= \left\langle \sum_{x'} y_{x'} x', x \right\rangle = \sum_{x'} y_{x'} K_{x,x'} = (Ky)_x \\
\langle g, x \rangle &= \langle Gg, x \rangle / G = (Ky)_x / \, ||y||_K
\end{aligned}$$

Each w_t is a joint sum g_t scaled to norm β_t. Taking $\beta_t = \beta \to 0$ and $T \to \infty$ yields a continuous limit as a dynamical system. The discrete iteration number $t > 0$ becomes a real-valued time. Taking $\frac{d}{d\beta} y \Big|_{\beta \to 0}$ yields the instantaneous change with time:

$$\frac{d}{dt} y_x = \frac{d}{d\beta} e^{-\beta |\langle g,x \rangle|} y_x \Bigg|_{\beta \to 0} = -\frac{|Ky|_x}{\sqrt{y^T K y}} y_x \tag{16}$$

The partial proof of conjecture analyzes this dynamical system.

4.3. Intuitions for the algorithm

Properly learning versus halfspaces means finding a vector that optimizes correlation:

$$v_1 = \underset{w \in \mathbb{R}^n}{\operatorname{argmax}} \ \chi(h_w^{0\text{-}1}) \tag{17}$$

We decompose this hard problem into finding a 'head' vector of norm $\beta_1 \in (0, 1]$, and subsequently competing with a 'tail' or 'completion' vector.

The smooth halfspace h_{v_1} is equivalent to the smooth list with head $\beta_1 v_1$ and tail $(1 - \beta_1) v_1$, for any $\beta_1 \in [0, 1]$ (theorem 10). Similarly, a halfspace is equivalent to the smooth list with head $\beta_1 v_1$ and tail $L v_1$ for $L \to \infty$. So, judiciously constructing a smooth list is at least as good as solving 17. A strictly better classifier could be obtained by picking vectors which point in different directions. Indulging in this flexibility could ostensibly lead to overfitting. However, in the worst case, the nonlinear power of smooth lists has essentially no cost: learning smooth lists does not require more data than learning (smooth) halfspaces (theorem 9).

The decomposition of fig. 5 involves maximizing two correlations:

- χ_1, which is identical to the original χ, and
- χ_2, which reweights the data according to whether the head element passed.

The reweighting may be interpreted in two ways: first as conditioning the marginal distribution of the inputs x, or second as an input-dependent scaling of the outputs y_x. We denote the first as as $\chi(\cdot \mid w_1 \text{ passes})$ and the second as χ_2 (c.f. fig. 4). The amount of

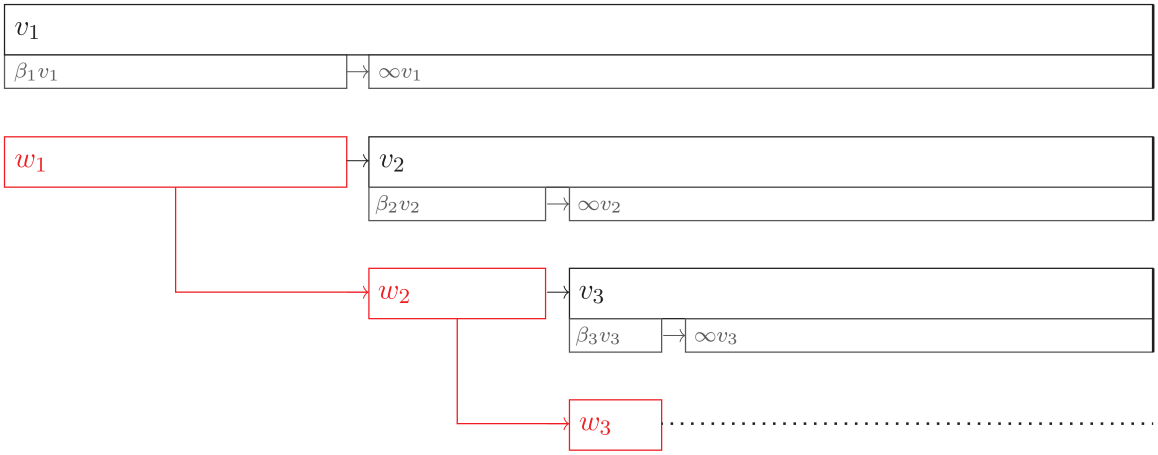

Figure 5: A visualization of the algorithm. The optimal smooth halfspace h_{v_1} decomposes into a smooth list of halfspaces consisting of the head $\beta_1 v_1$ and the tail Lv_1 for $L \to \infty$. These components are respectively bested by w_1, which is chosen by the algorithm, and v_2, which exists but is not known to the algorithm. The algorithm inductively competes against h_{v_2}, and produces the smooth list highlighted in red.

reweighting depends on just the input — in particular, on the confidence $|\langle w_t, x\rangle|$ of w_t on x — not whether it was correctly classified.

For small enough β_1, χ_1 and χ_2 may be independently maximized. No matter how the head w_1 is chosen, we conjecture there is always a tail v_2 which is as good as the competing tail v_1. Furthermore, the reweighting never causes the joint average to be zero, so the algorithm never gets stuck.

The key step is maximizing χ_1. This problem is nonconvex, but it has two pliancies: the solution just needs to match $h_{\beta_1 v_1}$, and β_1 is a variable. This new kind of problem is called β-competitive optimization. An upper bound on β_1 specifies a correct algorithm, and a lower bound on β_1 establishes its convergence rate. Our solution is intuitive: at the origin, the direction w_1 of instantaneous steepest ascent has an initial advantage over all other directions. It maintains its advantage over v_1 for some length β_1.

Quantifying β_1 involves a new analysis of agnostic learning. Even if the inputs are distributed simply, the outputs could be chosen by an adversary aiming to erode w_1's advantage within the shortest possible length. On rotationally invariant inputs, the adversary's choice reduces to picking the angle between w_1 and v_1. Since w_1 is just an average of the data, the angle relates to the correlation of v_1: small and large angles make $h_{\beta_1 v_1}$ have high and low correlation, respectively. In other words, the algorithm's choice of w_1 is a formal constraint on the adversary's strategy.

4.4. Experiments

Unlike previous agnostic learning algorithms, the experiments in this section show KG is empirically fast and practical, even in challenging scenarios. The experiments examine the

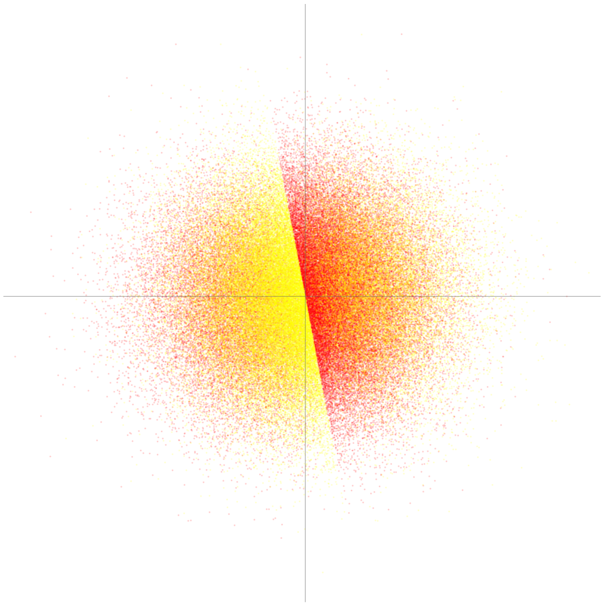

Figure 6: Example data for $n = 2$.

impact of unsound hypotheses and noisy data, compare the algorithm to the state of the art, and illustrate learning sparse parities with noise.

4.4.1. Sound hypotheses, noisy data

Even if the outputs are initially consistent with a hypothesis, noise foils relaxation. Reliably successful relaxation depends on fragile modeling assumptions or carefully preprocessed data. In this experiment (fig. 7), the inputs are standard normals in $\mathbb{R}^{128}$. The outputs are generated by a halfspace $h_w^{0\text{-}1}$, where $||w|| = 1$, and flipped with probability increasing with their unsigned margin (their distance from the decision boundary).

$$y_x = \begin{cases} \operatorname{sgn}(\langle w, x\rangle) & \text{with probability} \exp(-k\sqrt{n}\,|\langle w, x\rangle|) \\ -\operatorname{sgn}(\langle w, x\rangle) & \text{otherwise} \end{cases}$$

In high dimension, the overwhelming majority of points have small margin. The parameter k exponentially increases the flip probability. Such outputs are visualized in fig. 6. A simple integral (see the technical appendix) calculates:

$$\mathsf{opt}^{0\text{-}1} = -1 + 2\exp\left(\frac{k^2 n}{2}\right)\operatorname{erfc}\left(\frac{k\sqrt{n}}{\sqrt{2}}\right) \tag{18}$$

Relaxation cannot cope with higher levels of noise. Our algorithm converges to the calculated optimum in a pleasant meeting of theory and experiment.

4.4.2. Unsound hypotheses, no noise

Even if the outputs are deterministically generated from the inputs, relaxation can fail, and lifting can need too much time and data. In the second experiment (fig. 8), the inputs

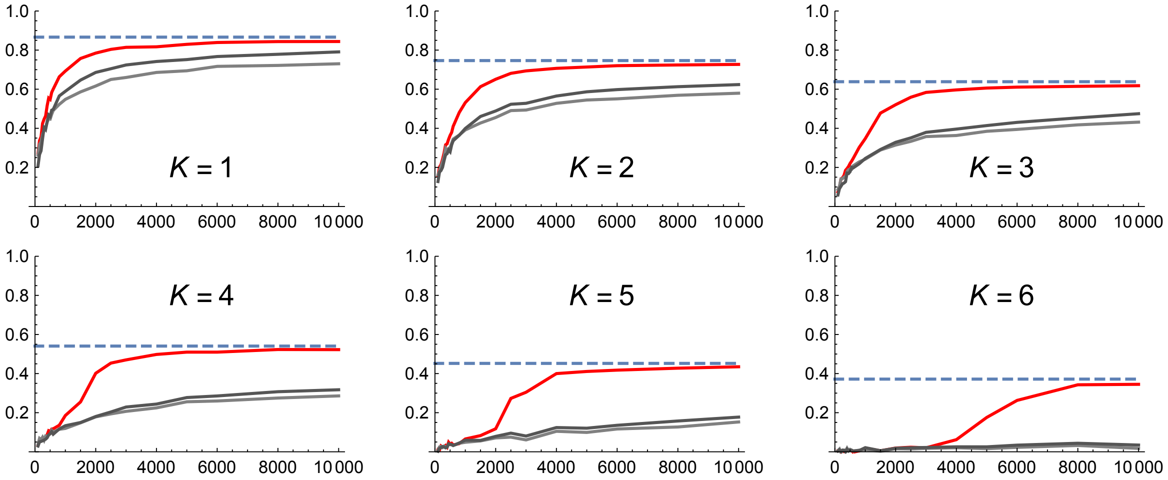

Figure 7: For varying noise levels, correlation varying with the amount of data. SVM (Pegasos), ℓ_2-regularized logistic regression (LIBLINEAR), and our algorithm, relative to $\mathsf{opt}^{0\text{-}1}$, the correlation of the optimal halfspace.

are standard normals of varying extrinsic dimension n. The outputs are generated by a parity function of fixed intrinsic dimension (sparsity) of $k = 3$. Relaxation is no better than random guessing. Kernel SVM with a degree-$(k + 1)$ polynomial is reliable because it subsumes the degree-k parity function. However, as the extrinsic dimension increases, it requires an overwhelming amount of data. Our algorithm uses a modicum of data and reliably achieves a nontrivial correlation.

5. Normally distributed inputs and consequences

Conjecture 11 is unsatisfactory because it does not bound the amount of time used by the algorithm. We propose to prove such a bound when the inputs are normally distributed. To summarize the discussion of section 2, the state of the art for this problem is:

- versus smooth halfspaces, sampling uses $m = O(L^2/\epsilon^2)$ data and $\mathrm{poly}(m, n, 2^{O(1/\epsilon^2)})$ time.
- versus halfspaces, lifting uses $m = \mathrm{poly}\left(n^{\frac{\log^3(1/\alpha)}{\alpha^2}}, \log(\frac{1}{\epsilon})\right)$ data and $\mathrm{poly}(m, n, 1/\epsilon)$ time (Daniely, 2014).

We believe it is possible that KG uses polynomial time in this scenario.

Conjecture 12 *With:*

- *the standard normal distribution (with n iid $N(0, 1)$ coordinates) on inputs $x \in \mathbb{R}^n$,*
- *bounded outputs $|y_x| \leq 1$,*

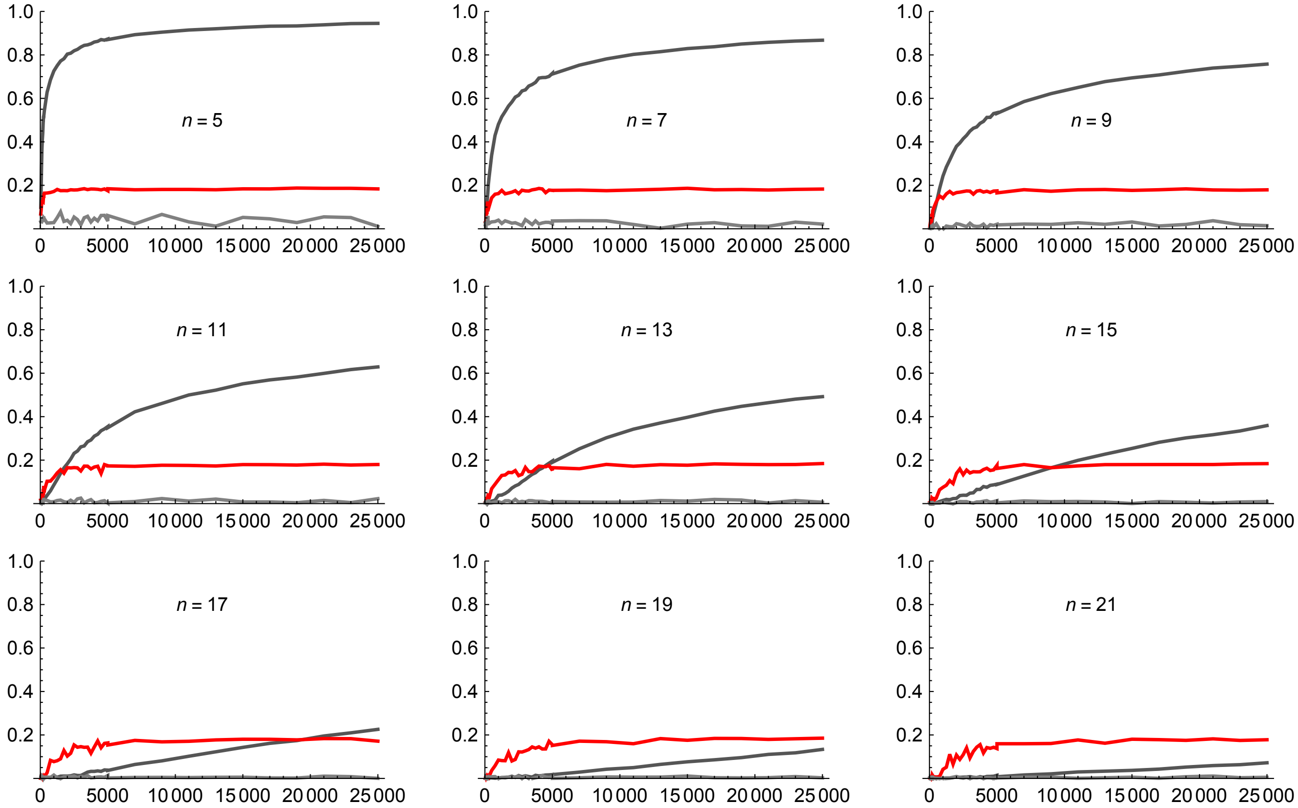

Figure 8: Correlation varying with the amount of data. SVM (LIBLINEAR), degree-4 polynomial SVM (LIBSVM), and our algorithm.

- *a nonzero joint average* $\mathop{\mathbf{E}}_{x \sim \mathcal{D}} (x \cdot y_x)$*,*
- $\mathit{opt}^{0\text{-}1}$ *defined relative to halfspaces, per eq.* (1)*, whose vectors* w *have integer components and satisfy* $||w|| \leq W$*.*
- *error parameters* $\alpha, \epsilon \in (0, 1)$*,*
- *an appropriate setting of* β_t *for* $T = O(\frac{1-\alpha}{\alpha} \cdot \frac{W}{\epsilon(\mathit{opt}^{0\text{-}1} - \epsilon/2)})$ *iterations,*

the algorithm uses $m = O(n/\epsilon^2)$ *data and* $O(m \cdot n \cdot T)$ *time to agnostically learn halfspaces, i.e. produce* c *satisfying* $\chi(c) \geq (1 - \alpha)\mathit{opt}^{0\text{-}1} - \epsilon$*.*

This does not contradict any of the hardness results described in section 2. However, it does have consequences for the problem of learning sparse parities with noise, which we now describe.

Let S be an unknown, size-k subset of $\{1, \ldots, n\}$. Consider recovering S from this distribution:

- the input s is distributed uniformly on the hypercube $\{-1,1\}^n$,
- the binary output is the parity on S, i.e. $\prod_{i \in S} s_i$, negated with probability $\eta \in [0, 1/2)$.

When $\eta = 0$, recovery is as easy as solving linear equations. When $\eta > 0$, it is notoriously challenging, and is called learning sparse parities with noise. It is closely related to fundamental problems in a variety of disciplines.

- cryptography: learning (non-sparse) parities is a common cryptographic hardness assumption (Pietrzak, 2012).
- coding theory: learning (sparse) parities with noise is equivalent to decoding random (sparse) linear codes (Blum et al., 2003). Such decoding comprises the best-known attack against the McEliece cryptosystem, the most promising approach to post-quantum cryptography (Becker et al., 2012).
- combinatorial optimization: given a system of inconsistent linear equations, each involving just k variables, MAX-k-LIN-2 is the problem of satisfying as many such equations as possible. It involves sparse data rather than a sparse underlying assignment, a dual assumption (Applebaum et al. (2010) footnote 6). It is NP-hard to approximate and underlies many further hardness results (Håstad, 2001).

A brute-force algorithm for recovering S uses $O(n^k)$ time. The best known algorithm is only a slight improvement: it uses $O\left(n^{0.8k}\text{poly}\left(\frac{1}{1-2\eta}\right)\right)$ time (Valiant, 2012) (see also (Karppa et al., 2016)). Currently, learning sparse parities with noise is "widely believed to be intractable" (Feldman et al., 2013) and "even an $O(n^{k/2})$-time algorithm for LSPN would be considered a breakthrough result" (Klivans and Kothari, 2014). Under conjecture 12, the KG algorithm, paired with the reduction of (Klivans and Kothari, 2014), would use polynomial time even when $k = O(\log n)$.

Conjecture 13 *Under conjecture 12, learning sparse parities with noise uses* $\tilde{O}\left(\frac{2^{c \cdot k} n^4 \sqrt{n}}{(1-2\eta)^4}\right)$ *time and data for some constant c.*

This would be fast enough to solve problems tied to the foundations of learning. The probably approximately correct model is the first, enduring formalization of learning a function from data (Valiant, 1984). In 1984, when introducing the model, Valiant considered learning boolean formulae, with ℓ terms in disjunctive normal form, from inputs distributed uniformly on the hypercube. He described it as 'tantalizing' and 'apparently simple.' It reduces to learning logarithmically-sparse parities with noise (Feldman et al., 2009).

Conjecture 14 *Let $\mathcal{D}$ have inputs distributed uniformly on the hypercube, and set* $\textsf{opt} = 1$ *relative to n-variable, ℓ-term DNFs. Under conjecture 12, obtaining c satisfying* $\chi(c) \geq \textsf{opt} - \epsilon$ *uses* $\text{poly}(n, \ell, 1/\epsilon)$ *time and data.*

The best previous algorithm uses $O(n^{\log(\ell/\epsilon)})$ time (Verbeurgt, 1990).

The statistical query model prevents learning algorithms from directly sampling data; it allows only imprecise, real-valued questions about the distribution (Kearns, 1993). (See appendix A.4 for a precise definition.) This restriction is theoretically mild: a polynomial number of queries can typically replace a polynomial amount of data. The main counterexample involves parities. In 1993, when introducing the model, Kearns noted parities cannot be learned using polynomial queries, even with unlimited time and $\mathsf{opt} = 1$. This still holds when $k = \log n$ (Blum et al., 1994), or if $S \subseteq \{1, \dots, \log n \log\log n\}$ (Blum et al., 2003). By contrast, learning the latter kind of parities directly from noisy data uses polynomial time (Blum et al., 2003). Since the statistical query model doesn't impede most existing algorithms, but does render some distributions unlearnable, new algorithms that don't fit within the model seem important.

Our algorithm fits the model. (Of course, the parity reduction does not.)

Conjecture 15 *Under conjecture 12 and the conditions of conjecture 11, learning versus smooth halfspaces uses* $\mathrm{poly}(n, L, 1/\alpha, 1/\epsilon, 1/\mathsf{opt})$ *time and statistical queries. Similarly, under the conditions of theorem 12, learning versus halfspaces uses* $\mathrm{poly}(n, \beta, 1/\alpha, 1/\epsilon, 1/\mathsf{opt}^{0\text{-}1})$ *time and statistical queries.*

Under the conjecture, KG would be the first algorithm which fits the statistical query model but, due to the parity reduction, is critically useful beyond the model's confines. It suggests the model is more inclusive than previously thought; perhaps it captures the essence of learning, and excludes pathologies which may be addressed separately. Statistical queries accommodate noise-tolerant, privacy-preserving, distributed, and evolutionary learning, so these endeavors may be more fruitful than previously imagined.

The algorithm would also fast enough to claim a \$1000 bounty posted in 2003 by Blum (Blum, 2003). It concerns k-juntas: boolean functions which depend on only $k < n$ variables.

Conjecture 16 *Under conjecture 12, learning versus* $\log(n)$*-juntas uses* $\mathrm{poly}(n)$ *time and data, given* $\mathsf{opt} = 1$ *and inputs distributed uniformly on the hypercube.*

The best known algorithm uses $O(n^{0.6 \log n} \mathrm{poly}(n))$ time (Valiant, 2012).

Though theorem 12 would settle many questions in learning theory, it would not compromise the security of any well-studied cryptographic schemes. In particular, the security of the schemes in (Pietrzak, 2012) depends crucially on non-sparse secrets. Section 7.1.2 examines the importance of sparsity.

6. Agnostic linear multiclassification

The second half of this proposal concerns conditional distributions supported on $q > 2$ elements. The outputs y_x are in the $q - 1$-simplex, i.e. a distribution over q elements. $c(x)$ is a distribution conditioned on the input $x \in \mathbb{R}^n$. The correlation of c is:

$$\chi(c) = \mathop{\mathbf{E}}_{x} \left(\langle c(x), y_x \rangle\right)$$

This may be rephrased as classification to $q > 2$ classes. (For brevity, we call this multiclassification, and still call $q = 2$ classification.) Identify e_j, the point distribution on output j, with the integer j. This maps a conditional distribution over vectors y_x to a distribution over integers j_x.

$$\chi(c) = 1 - 2\mathop{\mathbf{P}}_{x} \left(\mathop{\mathbf{P}}_{j \sim c(x)} (j \neq j_x) \right)$$

Similarly redefine the margin correlation for a conditional distribution. This is simply that the output produced for the correct class j is at least γ larger than all the other outputs.

$$\chi_\gamma(f) = \underset{x}{\mathbf{P}} \left(\forall j \neq j_x, f(x)_{j_x} - f(x)_j \geq \gamma \right) \tag{19}$$

6.1. Learning versus Voronoi diagrams

A Voronoi diagram is a multiclassifier which identifies each class $j \in [q]$ with a vector W_j. Given an input x, it returns the closest class:

$$h_W^{0\text{-}1}(x) = \underset{j}{\text{argmin}} \; ||W_j - x||^2 = \underset{j}{\text{argmax}} \; \langle W_j, x \rangle$$

The last equality assumes the vectors W_j have equal Euclidean norm. If they don't, the multiclassifier is sometimes referred to as the 'multivector construction', or simply the functions returned by multiclass support vector machines.

The amount of data needed to learn Voronoi diagrams is bounded in terms of n and q. The following theorem generalizes theorem 1. It is based on a multiclass generalization of VC dimension called Natarajan dimension, which is nq for Voronoi diagrams.

Theorem 17 *If* $m \geq O(\frac{nq \log(nq) \log(q)}{\epsilon^2} + \log(1/\delta))$, *then*

$$\sup_{W \in \mathbb{R}^{q \times n}} \chi(h_W) - \hat{\chi}(h_W) \leq \epsilon$$

with probability $1 - \delta$ *((Daniely et al., 2013) lemma 15 and theorem 22).*

Rademacher complexity controls the margin correlation of a Voronoi diagram. The following theorem essentially generalizes theorem 3 and theorem 4. (We shall define smooth Voronoi diagrams, and anticipate a relationship between margin correlation of a Voronoi diagram and correlation of a smooth Voronoi diagram.)

Theorem 18 *Let the inputs* $||x|| \leq 1$ *and let* $||W||_{2,2} \leq \beta$. *Then:*

$$\sup_W \chi_\gamma(h_W) - \hat{\chi_\gamma}(h_W) \geq \frac{q}{\gamma}\sqrt{\beta}m + \frac{\delta}{\sqrt{m}}$$

with probability at most $e^{-2\delta^2}$ *(Maximov and Reshetova (2015) theorem 1).*

Recent work suggests a sublinear dependence on q (Lei et al., 2015).

6.2. Previous algorithms

We are not aware of any agnostic multiclassification algorithms besides brute-force search. In practice, it is popular to reduce multiclassification to classification in the following ways.

- Error-correcting output codes form Q binary classification problems by partitioning the classes (Dietterich and Bakiri, 1995). Each class q is identified with a binary code vector of length Q. Given a new input, running each classifier produces a binary vector of length Q, which is then multiclassified according to the closest code vector. For example, one-versus-rest involves q classification problems between each class and all the others. One-versus-one (or 'all-pairs') involves $q(q-1)/2$ problems between each pair of classes.

- A tree multiclassifier identifies each leaf of a binary tree with a different class, and employs a classifier at all the internal nodes. It applies the root classifier to the input, recurses along the left or right subtree if the output is respectively negative or positive, eventually reaches a leaf, and finally returns the associated class.

Such reductions may generate difficult classification problems; these thwart absolute reductions, which guarantee high multiclass correlation if each of the binary correlations is high. Fortunately, there are relative reductions which guarantee high multiclass correlation if each of the binary correlations is relatively high (Beygelzimer et al., 2016). For example, there are two approaches to training tree classifiers:

- top-down: start at the root and split the data according to the classes of the left subtree and those of the right. Train a classifier separating the left and right data. Finally, recurse upon the left and right subtrees with the left and right data, respectively.

- bottom-up: associate each leaf node with data of the same class. At each node directly above the leaves, train a classifier separating the left and right data. Associate correctly classified data with the node. Recurse until the root is trained.

The bottom-up approach has a relative guarantee, whereas the top-down approach does not. Unfortunately, such reductions do not preserve agnostic learning guarantees due to the definition of 'relative'. The reduction involves learning versus all possible classifiers, not just over the hypotheses. High correlation is defined relative to the Bayes-optimal correlation. There is no guarantee that opt (that is, the best hypothesis) is close to this.

Another problem with these reductions is that a code choice or tree shape implicitly encodes prior knowledge about which classes are separable from one another. In the absence of such prior knowledge, a random code or tree is typically used. Unfortunately, if $n \ll q$, then with high probability over the code or tree, any resulting multiclassifier has close to 0 correlation (Daniely et al., 2012).

By contrast, (Daniely et al., 2012) is especially supportive of Voronoi diagrams. Voronoi diagrams, tree multiclassifiers, and one-versus-all have the same Natarajan dimension; in an asymptotic, worst-case sense, they all use the require the same amount of data. However, opt is always at least as high for Voronoi diagrams as for the other multiclassifiers, and is sometimes strictly higher.

7. Proposed generalizations

This section generalizes smooth lists of halfspaces and the KG algorithm to $q > 2$. A smooth Voronoi diagram is a randomized multiclassifier defined by a matrix $W \in \mathbb{R}^{q \times n}$ whose rows W_j have the same norm. Given an input x, it operates as follows:

```
Activate each coordinate $j \in \{1, \ldots, q\}$ with probability $1 - e^{-L|\langle W_j, x\rangle|}$.
If multiple coordinates activate, repeat.
If exactly one coordinate $j$ activates, return $e_j$.
If no coordinates activate, return $e_j$ uniformly at random.
```

A smooth list of Voronoi diagrams is defined by a list of matrices $W_1, \ldots, W_T$. Given an input x, it operates as follows:

```
For $t = 1, \dots, T$:
  For $j \in \{1, \dots, q\}$
    $W_{t,j} = \frac{1}{m} \sum_{i=1}^{m} x_i y_{x_i,j}$
    $W_{t,j} = \beta_t W_{t,j} / \, ||W_{t,j}||$
    $y_{x_i} = \prod_{j=1}^{k} e^{-L|W_{t,j}|x_i} y_{x_i}$
Return $W_1, \dots, W_T$
```

```
For $t = 1, \dots, T$:
  With probability $1 - e^{-|\langle w_t, x \rangle|}$, return $\mathrm{sgn}(\langle w_t, x \rangle)$
Return $-1$ or $1$ uniformly at random.
```

Figure 9: The proposed training algorithm (left) and multiclassification algorithm (right).

```
For $t = 1, \dots, T$:
  Activate each coordinate $j \in \{1, \dots, q\}$ with probability $1 - e^{-L|\langle W_{t,j}, x \rangle|}$.
  If multiple coordinates activate, repeat.
  If exactly one coordinate $j$ activates, return $e_j$.
  If no coordinates activate, continue.
Return $e_j$ uniformly at random.
```

The algorithm in figure 9 naturally generalizes the one in section 4.2. We postulate that it is a correct agnostic learning algorithm.

Conjecture 19 *Let $\mathcal{D}$ be any distribution on bounded inputs $||x|| \leq 1$ and outputs y in the $q-1$-simplex. Define opt relative to smooth halfspaces, per eq. (6), with slope L and vectors $||w|| \leq 1$. For any $\epsilon \in (0,1)$, the algorithm uses $m = O(L^2 q^2/\epsilon^2)$ data and $O(m \cdot n \cdot q \cdot T)$ time. For some positive sequence $\{\beta_t\}_{t=1}^{T}$, the resulting classifier c satisfies $\chi(c) \geq \mathsf{opt} - \epsilon$ for sufficiently large T.*

7.1. The hardness of multiclassification

The multiclass generalization of KG may be fast in practice. However, we do not propose to meaningfully bound the amount of time it takes. We do not believe fast agnostic multiclassification is achievable by any algorithm, even under restrictions similar to those in section 5. The difficulty moving from $q = 2$ to $q > 2$ echoes the hardness encountered when moving beyond quadratic forms in many areas of computer science. For example:

- semidefinite programming and polynomial programming (for optimizing over positive, degree-q polynomials),
- linear and multilinear algebra (for computing the rank, nuclear norm, eigenvalues, eigenvectors, etc. of a degree-q tensor.)

Surprisingly, for multiclassification, current hardness results for $q > 2$ are no stronger than for $q = 2$; they do not exploit growing q. Furthermore, the hardness of multiclassification on normally distributed inputs is not understood. Its relationship to learning with errors with modulus q, the multiclass generalization of learning parities, is not understood, particularly when the unknown vector is k-sparse. We propose to fill these gaps by formalizing connections between the following problems.

- Section 7.1.1 describes how min-sum clustering, a problem that is likely NP-hard to approximate, reduces to multiclassification with statistical queries. This suggests that multiclassification is harder than classification in the worst case.
- Section 7.1.2 describes why the distinction between sparse and non-sparse parities, which underlies the optimism of theorem 12, is irrelevant when $q > 2$. This suggests that, even when the inputs are normally distributed, multiclassification remains hard.

7.1.1. Min-sum clustering

A clustering assigns each datum $x_1, \dots, x_m$ to one of q clusters. An optimal min-sum clustering minimizes the sum of intracluster distances according to a metric $d(\cdot, \cdot)$. That is, the clustering $c = C_1, \dots, C_k$ minimizes the following objective:

$$\kappa(c) = \frac{1}{m} \sum_{j \in [q]} \sum_{x, x' \in C_j} d(x, x')$$

(For example, consider assigning k tables to m people while minimizing mutual discord.) If $d(\cdot, \cdot)$ is the squared Euclidean distance (which is not a metric, but is nonetheless algebraically convenient), then κ is equivalent to balanced q-means clustering (e.g. Awasthi and Balcan (2014) corollary 3):

$$\kappa(c) = \sum_{j \in [q]} \sum_{x, x' \in C_j} \left|\left|x - x'\right|\right|^2 = \sum_{j \in [q]} 2\left|C_j\right| \sum_{x \in C_j} ||U_j - x||^2 = \kappa(U)$$

Each U_j is the average of cluster j. This is a center-based clustering because there is a one-to-one relationship between the clustering c and the means U_j. This problem is not as well-studied as standard q-means clustering, which omits the cluster sizes. That is, the q-means objective is:

$$K(U) = \sum_{j \in [q]} \sum_{x \in C_j} ||U_j - x||^2$$

Here is some motivation for the more obscure balanced problem. In practice, imbalanced clusters may be undesirable; in the aforementioned example, the tables should be of comparable size. In theory, q-means clustering is known to be hard to approximate (Lee et al., 2015; Awasthi et al., 2015b), but no hardness results are known for the balanced variant. We fill this gap by showing balancing preserves hardness. We conjecture balanced q-means is hard to approximate to a constant factor.

Conjecture 20 *For data $x_1, \dots, x_m \in \mathbb{R}^n$, let c^* be the best min-sum clustering, i.e. the minimizer of κ. If $\mathrm{P} \neq \mathrm{NP}$, for some $\epsilon > 0$, finding c satisfying $\kappa(c) \leq (1 + \epsilon)\kappa(c^*)$ is not possible with $\mathrm{poly}(m, n, q)$ time.*

Min-sum clusterings may be found by any classification algorithm which accesses data solely to evaluate its performance – an interface known as 'correlational statistical queries' or a 'zero-order oracle'. (Recall section 5.) We expect the following reduction.

Conjecture 21 *For every min-sum clustering objective κ defined by data $x_1, \dots, x_m$, there is a classification correlation χ such that $\kappa(c) = \chi(c)$. On the right hand, c denotes a classifier; on the left hand side, it denotes the clustering obtained by evaluating the classifier on the data.*

7.1.2. Learning with errors

Learning with errors (LWE) generalizes learning parities with noise to $q > 2$ (Regev, 2010). The problem parameters are the dimension n, the sparsity k, the standard deviation of error $\eta > 0$, and the amount of data m. The secret s is an unknown, k-sparse vector in $\mathbb{Z}_q^n = \{-q/2, q/2\}^n$ which generates the following joint distribution on inputs and outputs:

- the inputs x are distributed uniformly on $\mathbb{Z}_q^n$,
- the outputs $y_x = \langle s, x\rangle + \xi$. The error ξ is a discrete Gaussian, a random variable supported on $\mathbb{Z}_q^n$ whose probability mass is proportional to $e^{-\pi||\xi/\eta||^2}$.

Given m data drawn from this distribution, search-LWE is recovering s, and decision-LWE is distinguishing the data from uniformly random values. The relationship between LWE and agnostic multiclassification remains the same between $q = 2$ and $q > 2$:

- decision-LWE with an arbitrary secret reduces to decision-LWE with a uniformly random secret in $\mathbb{Z}_q^n$. This randomized self-reduction allows any learning algorithm which succeeds with nontrivial probability to be boosted by repetition.
- search-LWE reduces to decision-LWE (albeit with increased error) by testing whether each secret coordinate s_i is nonzero (Micciancio and Peikert (2011) theorem 3.1; see appendix A.3 for $q = 2$.) Dropping a nonzero coordinate completely decorrelates the inputs and outputs, so each test may be implemented as a learning problem: if nonzero correlation is still achievable after dropping the coordinate, then it is zero.

In analogy to theorem 25 for $q = 2$, it is reasonable to believe that Voronoi diagrams are inversely-polynomially-correlated with log-sparse secrets, in the following sense.

Conjecture 22 *There is a map $\phi : \mathbb{Z}_q^n \to \mathbb{R}^N$ from the uniform distribution to the normal distribution such that, for any k-sparse $s \in \mathbb{Z}_q^n$,*

$$\max_{W\in\mathbb{R}^{q\times N}} \mathop{\mathbf{P}}_{x\sim\mathbb{Z}_q^n} \left(h_W(\phi(x)) = \langle s, x\rangle\right) = \Omega(1/\mathrm{poly}(n, q, \log k))$$

The parity-preserving transformation of (Brakerski et al., 2013) may help prove this conjecture.

The security of much modern cryptography — including schemes for fully homomorphic encryption, functional encryption, identity-based cryptography, and leakage-resilient cryptography — depends on the difficulty of decision-LWE. It is an appealing foundation for cryptography because solving random instances reduces to solving worst-case instances of fundamental lattice problems (Brakerski et al., 2013). LWE is widely believed to require $2^{\Omega(n)}$ time with appropriate choices for the parameters. We are presently concerned with $q > 2$ versus $q = 2$, as well as the sparsity k of s; it seems $q \gg 2$ is essential, at which point k is irrelevant. The importance of large q has been understood since the seminal paper introducing LWE:

> "It seems that in order to prove similar results for smaller values of q, substantially new ideas are required. Alternatively, one can interpret our inability to prove hardness for small q as an indication that the problem might be easier than believed." (Regev, 2009)

q and η should grow as polynomials of n, otherwise two devastating attacks are possible:

- a 'structured noise' attack takes $O(\exp(q \cdot \eta)^2)$ time, which is subexponential if $q \cdot \eta < \sqrt{n}$ (Arora and Ge, 2011). (Curiously, this attack does not apply when $q = 2$.)
- if q is exponential in n but η is polynomial, then search-LWE may be solved in polynomial time (Laine and Lauter, 2015).

A technique called modulus switching reduces q by increasing n or η, and effectively characterizes the hardness of LWE by $n \log_2 q$ rather than by each parameter individually (Brakerski et al., 2013). However, it cannot reduce to $q = 2$.

Sparsity is not as thoroughly studied; there is no formal reduction from large k to small k. For numerical reasons, various implementations of fully homomorphic encryption employ small k. (Somewhat tangentially: the difficulty of a sparse variant of subset sum underpinned such schemes, but this dependence has been lifted (Brakerski and Vaikuntanathan, 2014).) (Gentry et al. (2012) section C.1.1) informally argues the best known exploit of small k is not quantitatively advantageous. It is possible to transform LWE instances by swapping parts of the secret and error. Since small k implies small norm — sparse secrets are short — this transformation yields smaller η. The ineffectiveness of this attack is not surprising; modulus switching justifies the use of a short secret, drawn uniformly in $\{-1, 1\}^n$ rather than $\mathbb{Z}_q^n$, so long as the dimension is moderately increased from n to $n \log_2 q$ (Brakerski et al. (2013), theorem 4.1). Similarly short error vectors are admissible if m is only linear in n (Micciancio and Peikert, 2013).

Here are our own intuitions about why sparsity is irrelevant.

Conjecture 23 *For every $s \in \mathbb{Z}_q^n$, there is a $k = O(\log n)$-sparse $z \in \mathbb{Z}_q^n$ such that:*

$$\mathop{\mathbf{E}}_{x \sim \mathbb{Z}_q^n} (\langle s, x \rangle \langle z, x \rangle) = \Omega(1/\mathrm{poly}(n, q))$$

That is, log-sparse secrets are inversely-polynomially-correlated with non-sparse secrets.

Proof (intuition only): decision-LWE with a uniformly random secret reduces to decision-LWE with a secret drawn from a distribution with sufficient min-entropy (Goldwasser et al., 2010). The 'normal' form of LWE draws the secret and errors from the same distribution (Brakerski et al. (2013) lemma 2.12). Decompose a non-sparse discrete Gaussian secret $s \in \mathbb{Z}_q^n$ into z, the vector of its $\log(n)$-largest coordinates, and the orthogonal remainder r of small coordinates. We aim to show z is substantially correlated with s:

$$\mathop{\mathbf{E}}_{x \sim \mathbb{Z}_q^n} (\langle z, x \rangle \langle s, x \rangle) = \Omega(1/n)$$

An analysis based on the continuous Gaussian is justified since the standard deviation $\sqrt{n}$ is above the 'smoothing' value, at which the discrete Gaussian resembles the continuous Gaussian (Micciancio and Regev, 2007). A log-sparse secret captures more than $\frac{1}{n}$-fraction of the ℓ_1 norm of the whole secret. (Just compare the mean of $|s_i|$ with the integration of the PDF from τ to ∞, where $\mathbf{P}(|s_i| \geq \tau) = \log n/n$.) The desired correlation lower bound follows from the Pythagorean theorem. ■

The equivalence of sparse and non-sparse secrets does not extend to $q = 2$, where the discrete Gaussian is replaced by a Bernoulli random variable with expectation $\eta \in (0, 1/2)$. An error distribution producing log-sparse (on average) secrets has $\eta = \log n/n$. This is a tiny noise rate of standard deviation at most $\sqrt{\log n}$. The probability of drawing $O(n)$ noise-free samples becomes $O(1/n)$, which means Gaussian elimination succeeds in finding the secret with inverse polynomial probability. This somewhat counterintuitive property — that LWE is harder than LPN but somehow admits more structure — was also encountered by Arora and Ge, whose 'structured noise' attack on low-noise LWE which does not carry over to LPN.

Combining the previous correlation conjectures leads to the following (non-quantitative) conjecture.

Conjecture 24 *Agnostically learning Voronoi diagrams with normally distributed inputs is as hard as learning with errors.*

Understanding the concrete, non-asymptotic difficulty of LWE is very important (Albrecht et al., 2015; Peikert, 2016). It is a largely experimental endeavor; multiclassification experiments with our algorithm may elucidate this difficulty.

References

Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. *CoRR*, abs/1505.04383, 2015. URL http://arxiv.org/abs/1505.04383.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4): 343–370, 1988. ISSN 1573-0565. doi: 10.1023/A:1022873112823. URL http://dx.doi.org/10.1023/A:1022873112823.

Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.

Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jir Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011. ISBN 978-3-642-22005-0. URL http://dblp.uni-trier.de/db/conf/icalp/icalp2011-1.html#AroraG11.

P. Awasthi, M.-F. Balcan, N. Haghtalab, and R. Urner. Efficient learning of linear separators under bounded noise. In Grünwald et al. (2015), pages 167–190. URL http://jmlr.org/proceedings/papers/v40/Awasthi15b.html.

Pranjal Awasthi and Maria-Florina Balcan. Center based clustering: A foundational perspective. 2014.

Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. In David B. Shmoys, editor, *STOC*, pages 449–458. ACM, 2014. ISBN 978-1-4503-2710-7.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34, pages 754–767. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015b.

Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1193–1200. Society for Industrial and Applied Mathematics, 2009.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944944.

Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1+1=0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-29010-7. doi: 10.1007/978-3-642-29011-4_31. URL http://dx.doi.org/10.1007/978-3-642-29011-4_31.

Shai Ben-David and Hans Ulrich Simon. Efficient learning of linear perceptrons. In *Advances in Neural Information Processing Systems 13*, pages 189–195. MIT Press, 2000.

Shai Ben-David, Philip M Long, and Yishay Mansour. Agnostic boosting. In *Computational Learning Theory*, pages 507–516. Springer, 2001.

Shai Ben-david, David Loker, Nathan Srebro, and Karthik Sridharan. Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1863–1870, 2012.

Ido Ben-Eliezer, Shachar Lovett, and Ariel Yadin. Polynomial threshold functions: Structure, approximation and pseudorandomness. *CoRR*, abs/0911.3473, 2009.

A. Beygelzimer, H. Daum, J. Langford, and P. Mineiro. Learning reductions that really work. *Proceedings of the IEEE*, 104(1):136–147, Jan 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2494118.

Aharon Birnbaum and Shai Shalev-Shwartz. Learning halfspaces with the zero-one loss: Time-accuracy tradeoffs. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 935–943, 2012.

Eric Blais, Ryan ODonnell, and Karl Wimmer. Polynomial regression under arbitrary product distributions. *Machine Learning*, 80(2-3):273–294, 2010. ISSN 0885-6125. doi: 10.1007/s10994-010-5179-6. URL http://dx.doi.org/10.1007/s10994-010-5179-6.

A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1):35–52, 1998. ISSN 1432-0541. doi: 10.1007/PL00013833. URL http://dx.doi.org/10.1007/PL00013833.

Avrim Blum. Learning a function of r relevant variables. In Bernhard Schlkopf and ManfredK. Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 731–733. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40720-1. doi: 10.1007/978-3-540-45167-9_54. URL http://dx.doi.org/10.1007/978-3-540-45167-9_54.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael 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*, STOC '94, pages 253–262, New York, NY, USA, 1994. ACM. ISBN 0-89791-663-8. doi: 10.1145/195058.195147. URL http://doi.acm.org/10.1145/195058.195147.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003. ISSN 0004-5411. doi: 10.1145/792538.792543. URL http://doi.acm.org/10.1145/792538.792543.

Stephane Boucheron, Olivier Bousquet, and Gabor Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 11 2005. ISSN 1262-3318. doi: 10.1051/ps:2005018. URL http://www.esaim-ps.org/article_S1292810005000182.

Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

Mark Bun and Thomas Steinke. Weighted Polynomial Approximations: Limits for Learning and Pseudorandomness. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 625–644, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-89-7. doi: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.625. URL http://drops.dagstuhl.de/opus/volltexte/2015/5327.

Tom Bylander. Learning noisy linear threshold functions. Technical report, 1998.

Shang-Tse Chen, Maria-Florina Balcan, and Duen Horng Chau. Communication efficient distributed agnostic boosting. *CoRR*, abs/1506.06318, 2015. URL http://arxiv.org/abs/1506.06318.

Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K. Lee. Submodular functions are noise stable. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1586–1592. SIAM, 2012. URL http://dl.acm.org/citation.cfm?id=2095116.2095242.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *arXiv preprint arXiv:1412.0233*, 2014.

Andrew Cotter, Shai Shalev-Shwartz, and Nathan Srebro. The kernelized stochastic batch perceptron. In *ICML*. icml.cc / Omnipress, 2012. URL http://dblp.uni-trier.de/db/conf/icml/icml2012.html#CotterSS12.

Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 498–511. SIAM, 2015. URL http://dl.acm.org/citation.cfm?id=2722129.2722163.

A. Daniely. A PTAS for Agnostically Learning Halfspaces. *ArXiv e-prints*, October 2014.

A. Daniely, N. Linial, and S. Shalev-Shwartz. The complexity of learning halfspaces using generalized linear methods. *ArXiv e-prints*, November 2012.

A. Daniely, N. Linial, and S. Shalev-Shwartz. From average case complexity to improper learning complexity. *ArXiv e-prints*, November 2013.

Amit Daniely. Complexity theoretic limitations on learning halfspaces. *CoRR*, abs/1505.05800, 2015. URL http://arxiv.org/abs/1505.05800.

Amit Daniely, Sivan Sabato, and Shai S Shwartz. Multiclass learning approaches: A theoretical comparison with implications. In *Advances in Neural Information Processing Systems*, pages 485–493, 2012.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *CoRR*, abs/1308.2893, 2013. URL http://arxiv.org/abs/1308.2893.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2933–2941. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5486-identifying-and-attacking-the-saddle-point-problem-in-high-dimensional-non-convex-o
pdf.

Ilias Diakonikolas, Prahladh Harsha, Adam Klivans, Raghu Meka, Prasad Raghavendra, Rocco A. Servedio, and Li-Yang Tan. Bounding the average sensitivity and noise sensitivity of polynomial threshold functions. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 533–542, New York, NY, USA,

2010. ACM. ISBN 978-1-4503-0050-6. doi: 10.1145/1806689.1806763. URL http://doi.acm.org/10.1145/1806689.1806763.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, pages 263–286, 1995.

Carlos Domingo and Osamu Watanabe. Madaboost: A modification of adaboost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, COLT '00, pages 180–189, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-703-X. URL http://dl.acm.org/citation.cfm?id=648299.755176.

V. Feldman, C. Guzman, and S. Vempala. Statistical Query Algorithms for Stochastic Convex Optimization. *ArXiv e-prints*, December 2015.

Vitaly Feldman. Evolvability from learning algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 619–628. ACM, 2008.

Vitaly Feldman and Pravesh Kothari. Agnostic learning of disjunctions on symmetric distributions. *CoRR*, abs/1405.6791, 2014. URL http://arxiv.org/abs/1405.6791.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, pages 711–740, 2013. URL http://jmlr.org/proceedings/papers/v30/Feldman13.html.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. ISSN 2050-1439. doi: 10.1111/j.1469-1809.1936.tb02137.x. URL http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine learning*, 43(3):293–318, 2001.

Craig Gentry, Shai Halevi, and Nigel P Smart. Homomorphic evaluation of the aes circuit. In *Advances in Cryptology–CRYPTO 2012*, pages 850–867. Springer, 2012.

Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *ICS*, pages 230–240. Tsinghua University Press, 2010. ISBN 978-7-302-21752-7. URL http://dblp.uni-trier.de/db/conf/innovations/innovations2010.html#GoldwasserKPV10.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Peter Grünwald, Elad Hazan, and Satyen Kale, editors. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Proceedings*, 2015. JMLR.org. URL http://jmlr.org/proceedings/papers/v40/.

A. Hanbo Li and J. Bradic. Boosting in the presence of outliers: adaptive classification with non-convex loss functions. *ArXiv e-prints*, October 2015.

Prahladh Harsha, Adam Klivans, and Raghu Meka. An invariance principle for polytopes. *J. ACM*, 59(6):29:1–29:25, January 2013. ISSN 0004-5411. doi: 10.1145/2395116.2395118. URL http://doi.acm.org/10.1145/2395116.2395118.

Prahladh Harsha, Adam Klivans, and Raghu Meka. Bounding the sensitivity of polynomial threshold functions. *Theory of Computing*, 10(1):1–26, 2014.

Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. ISSN 0004-5411. doi: 10.1145/502090.502098. URL http://doi.acm.org/10.1145/502090.502098.

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Lon Bottou, editors, *NIPS*, pages 793–800. Curran Associates, Inc., 2008. URL http://dblp.uni-trier.de/db/conf/nips/nips2008.html#KakadeST08.

Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. URL http://www.cs.mcgill.ca/~colt2009/papers/001.pdf#page=1.

Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008.

M. Kamrul Hasan and C. J. Pal. A New Smooth Approximation to the Zero One Loss with a Probabilistic Interpretation. *ArXiv e-prints*, November 2015.

Varun Kanade and Adam Kalai. Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems 22*, pages 880–888. 2009. URL http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips22/NIPS2009_0346.pdf.

Varun Kanade and Elchanan Mossel. MCMC learning. In Grünwald et al. (2015), pages 1101–1128. URL http://jmlr.org/proceedings/papers/v40/Kanade15.html.

Daniel M Kane, Adam Klivans, and Raghu Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. 30:522–545, 2013.

Matti Karppa, Petteri Kaski, and Jukka Kohonen. A faster subquadratic algorithm for finding outlier correlations. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1288–1305. SIAM, 2016. ISBN 978-1-61197-433-1. doi: 10.1137/1.9781611974331.ch90. URL http://dx.doi.org/10.1137/1.9781611974331.ch90.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 392–401, New York, NY, USA, 1993. ACM. ISBN 0-89791-591-7. doi: 10.1145/167088.167200. URL http://doi.acm.org/10.1145/167088.167200.

Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 341–352, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130424. URL http://doi.acm.org/10.1145/130385.130424.

Adam Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:63, 2014.

Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. *The Journal of Machine Learning Research*, 7:587–602, 2006.

Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning geometric concepts via gaussian surface area. In *FOCS*, pages 541–550. IEEE Computer Society, 2008. URL http://dblp.uni-trier.de/db/conf/focs/focs2008.html#KlivansOS08.

Adam R Klivans, Philip M Long, and Rocco A Servedio. Learning halfspaces with malicious noise. *The Journal of Machine Learning Research*, 10:2715–2740, 2009.

Kim Laine and Kristin E. Lauter. Key recovery for LWE in polynomial time. *IACR Cryptology ePrint Archive*, 2015:176, 2015. URL http://eprint.iacr.org/2015/176.

Quoc Le, Tamas Sarlos, and Alex Smola. Fastfood - approximating kernel expansions in loglinear time. In *30th International Conference on Machine Learning (ICML)*, 2013. URL http://jmlr.org/proceedings/papers/v28/le13.html.

Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *arXiv preprint arXiv:1509.00916*, 2015.

Yunwen Lei, Urun Dogan, Alexander Binder, and Marius Kloft. Multi-class svms: From tighter data-dependent generalization bounds to novel algorithms. In *Advances in Neural Information Processing Systems*, pages 2026–2034, 2015.

Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, July 1993. ISSN 0004-5411. doi: 10.1145/174130.174138. URL http://doi.acm.org/10.1145/174130.174138.

Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Mach. Learn.*, 78(3):287–304, March 2010. ISSN 0885-6125. doi: 10.1007/s10994-009-5165-z. URL http://dx.doi.org/10.1007/s10994-009-5165-z.

Philip M. Long and Rocco A. Servedio. Learning large-margin halfspaces with more malicious noise. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 91–99, 2011.

Y. Maximov and D. Reshetova. Tight Risk Bounds for Multi-Class Margin Classifiers. *ArXiv e-prints*, July 2015.

Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. Cryptology ePrint Archive, Report 2011/501, 2011.

Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *Advances in Cryptology–CRYPTO 2013*, pages 21–39. Springer, 2013.

Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

M.L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. Mit Press, 1972. ISBN 9780262130431. URL http://books.google.com/books?id=Ow1OAQAAIAAJ.

Cheong Hee Park and Haesun Park. A relationship between linear discriminant analysis and the generalized minimum squared error solution. *SIAM Journal on Matrix Analysis and Applications*, 27(2):474–492, 2005.

Chris Peikert. How (not) to instantiate ring-lwe. Technical report, 2016.

Krzysztof Pietrzak. Cryptography from learning parity with noise. In *SOFSEM 2012: Theory and Practice of Computer Science*, pages 99–114. Springer, 2012.

Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

Oded Regev. The learning with errors problem (invited survey). In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

Ronald L. Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

Frank Rosenblatt. The perceptron–a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003. ISBN 0137903952.

R Servedio and Leslie Valiant. Efficient algorithms in computational learning theory. *Harvard University, Cambridge, MA*, 2001.

Rocco A. Servedio. Smooth boosting and learning with malicious noise. *J. Mach. Learn. Res.*, 4:633–648, December 2003. ISSN 1532-4435. doi: 10.1162/153244304773936072. URL http://dx.doi.org/10.1162/153244304773936072.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

Shai Shalev-Shwartz and Yoram Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. *Machine learning*, 80 (2-3):141–163, 2010.

Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM J. Comput.*, 40(6):1623–1646, 2011.

Matus Telgarsky. A primal-dual convergence analysis of boosting. *Journal of Machine Learning Research*, 13:561–606, 2012. URL http://dl.acm.org/citation.cfm?id=2188405.

G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20, Oct 2012. doi: 10.1109/FOCS.2012.27.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL http://doi.acm.org/10.1145/1968.1972.

B. van Rooyen and A. Krishna Menon. An Average Classification Algorithm. *ArXiv e-prints*, June 2015.

Karsten Verbeurgt. Learning dnf under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, COLT '90, pages 314–326, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. ISBN 1-55860-146-5. URL http://dl.acm.org/citation.cfm?id=92571.92659.

Paul A. Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001. doi: 10.1109/CVPR.2001.990517.

Karl Wimmer. Agnostically learning under permutation invariant distributions. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 113–122. IEEE, 2010.

Tong Zhang. Regularized winnow methods. In *Advances in Neural Information Processing Systems*, pages 703–709, 2001.

Yuchen Zhang, Jason D. Lee, Martin J. Wainwright, and Michael I. Jordan. Learning halfspaces and neural networks with random initialization. *CoRR*, abs/1511.07948, 2015. URL http://arxiv.org/abs/1511.07948.

Appendix A. Proofs

A.1. Proof of theorem 9

Proof by induction on the list length T. In the base case $T = 2$, let $||w_1|| = \beta$ and $||w_2|| = 1-\beta$. Theorem 3 defines the Rademacher complexity $\mathcal{R}(\mathcal{F})$ as $\underset{\hat{\mathcal{D}},\sigma}{\mathbf{E}}\left(\hat{\mathcal{R}}(\mathcal{F})\right)$, the expectation of

```
For $i = 1, \ldots, n$:
  $f = KG(X_{-i}, Y, )$ trains without the $i$'th coordinate
  $c = \hat{\chi}(f)$ is an independent estimate of $\chi(f)$
  If $c$ is sufficiently large, keep coordinate $i$ in $S$.
```

Figure 10: The reduction of (Klivans and Kothari, 2014).

the empirical Rademacher complexity, in turn defined by the empirical distribution eq. (4). Recall the definition of smooth lists in eq. (15):

$$\begin{aligned}
\hat{\mathcal{R}}(\mathcal{F}) &= \sup_{w_1, w_2} \hat{\mathbf{E}}_x \left(h_{w_1}(x)\sigma_x + (1 - |h_{w_1}(x)|) h_{w_2}(x)\sigma_x\right) \\
&\leq \sup_{w_1} \hat{\mathbf{E}}_x \left(h_{w_1}(x)\sigma_x\right) + \sup_{w_1, w_2} \hat{\mathbf{E}}_x \left((1 - |h_{w_1}(x)|) h_{w_2}(x)\sigma_x\right) \\
&= \sup_{w_1} \hat{\mathbf{E}}_x \left(h_{w_1}(x)\sigma_x\right) + \sup_{w_1, w_2} \hat{\mathbf{P}}\left(w_1 \text{ passes}\right) \hat{\mathbf{E}}_x \left(h_{w_2}(x)\sigma_x \mid w_1 \text{ passes}\right) \\
&\leq \sup_{w_1} \hat{\mathbf{E}}_x \left(h_{w_1}(x)\sigma_x\right) + \sup_{w_1, w_2} \hat{\mathbf{E}}_x \left(h_{w_2}(x)\sigma_x \mid w_1 \text{ passes}\right)
\end{aligned}$$

The first equality takes the supremum over parameter vectors. The second inequality is the triangle inequality. w_1 affects the second correlation only through reweighting of the outputs, which is equivalently written in terms of conditional expectation. Conditioning the probability of input x on the event 'w_1 passes' means multiplying the probability of x by $1 - |h_{w_1}(x)|$, the probability that w_1 passes on x, and normalizing by the overall pass probability $\hat{\mathbf{E}}_x (1 - |h_{w_1}(x)|) = \hat{\mathbf{P}}(w_1 \text{ passes})$. The final inequality drops the pass probability of at most 1. Apply the worst-case bound theorem 4 to the different distributions to bound the Rademacher complexity:

$$\mathcal{R}(\mathcal{F}) \leq \mathcal{R}_\beta + \mathcal{R}_{1-\beta} = \mathcal{R}_1$$

A.2. Proof of theorem 10

Assume $a = \langle w, x \rangle \geq 0$; the case $a < 0$ is symmetric.

$$\begin{aligned}
f_{\beta w, (1-\beta) w}(x) &= (1 - e^{-\beta a}) + (1 - (1 - e^{-\beta a}))(1 - e^{-(1-\beta)a}) \\
&= 1 - e^{-\beta a} + e^{-\beta a}(1 - e^{-(1-\beta)a}) \\
&= 1 - e^{-\beta a - (1-\beta)a} \\
&= h_w(x)
\end{aligned}$$

A.3. Proofs of theorems 13, 14, and 16

The algorithm for learning sparse parities with noise uses the reduction of (Klivans and Kothari, 2014). Their reduction separately determines if each coordinate $i \in \{1, \ldots, n\}$ belongs to S. If $i \notin S$, dropping the i'th coordinate s_i of the input has no effect on the outputs. If $i \in S$, the outputs become uncorrelated with the inputs. If a classifier achieves nontrivial correlation when s_i is dropped, then $i \notin S$. Suppose a weak learner produces such a classifier whenever possible. To distinguish between the two cases, learn a classifier and estimate its correlation on a separate sample of data. The potential correlation is close to zero, so a large amount of data is required.

Weak learning can be implemented by learning versus halfspaces on standard normal inputs. Map the original distribution – with boolean inputs and noisy parity outputs – to a new distribution $\mathcal{D}$ with standard normal inputs and a nontrivial value of $\mathsf{opt}^{0\text{-}1}$ relative to halfspaces.

- Map the boolean input s to a standard normal input x by multiplying each coordinate with a half-normal random variable. Note $s_i = \operatorname{sgn}(x_i)$.
- Under this mapping, the parity is nontrivially correlated with a halfspace having small integer components. Their lemma 6:

  Lemma 25 *For every odd k,*

  $$\max_{w\in\{0,1\}^n} \mathop{\mathbf{E}}_{x}\left(h_w^{0\text{-}1}(x)\left(\prod_{i\in S}\operatorname{sgn}(x_i)\right)\right) \geq 2^{-\Theta(k)}$$

  The maximum is taken over 'majority' halfspaces. Note $||w|| \leq \sqrt{n}$.
- Map the outputs to $y_x = y_{\operatorname{sgn}(x)}$. If c has correlation θ with the parity on the new inputs, then its correlation with the new outputs is

  $$\chi(c) = (1-\eta)\theta + \eta(-\theta) = (1-2\eta)\theta.$$

With $\mathsf{opt}^{0\text{-}1} \geq (1-2\eta)2^{\Theta(-k)}$, the reduction is summarized by their lemma 5:

Theorem 26 *Let $\varepsilon = \mathsf{opt}^{0\text{-}1}/2$. If obtaining c satisfying $\chi(c) \geq \mathsf{opt}^{0\text{-}1} - \varepsilon$ uses τ time and data, then learning sparse parities with noise uses $\tilde{O}(n/\mathsf{opt}^{0\text{-}1}) + \tilde{O}(n)\cdot\tau$ time and data [1].*

?? obtains $\chi(c) \geq (1-\alpha)\mathsf{opt}^{0\text{-}1} - \epsilon$. Divide ε equally:

$$\varepsilon = \underbrace{\alpha\cdot\mathsf{opt}^{0\text{-}1}}_{\varepsilon/2} + \underbrace{\epsilon}_{\varepsilon/2}$$

Apply **??** with $\mathsf{opt}^{0\text{-}1} = (1-2\eta)2^{-\Theta(k)}$, $W = \sqrt{n}$, $\alpha = 1/4$, and $\epsilon = (1-2\eta)2^{-\Theta(k)}/4$ to obtain:

$$\tau \leq O\left(\frac{2^{\Theta(k)}n^{7/2}}{(1-2\eta)^4}\right)$$

Combine with theorem 26 and drop lower-order terms to prove theorem 13.

Theorem 14 and theorem 16 directly use the reductions of (Feldman et al., 2009). Their theorems 2 and 3:

Lemma 27 *Suppose learning sparse parities with noise uses $\tau(n, k, \frac{1}{1-2\eta})$ time and data. Then:*

[1] This mildly rephrases absolute error in terms of correlation: $\mathop{\mathbf{E}}_{x,y}(|c(x)-y|) = 2\mathop{\mathbf{P}}_{x,y}(c(x)\neq y) = 1-\chi(c)$.

- *under the conditions of theorem 14, learning length-ℓ DNFs uses $\tilde{O}(\frac{\ell^4}{\epsilon^2} \cdot \tau(n, \log B, B)^3)$ time, where $B = \tilde{O}(\ell/\epsilon)$.*
- *under the conditions of theorem 16, learning k-juntas uses $O(2^k k \cdot \tau(n, k, 2^{k-1})$ time.*

With theorem 13, obtain these times for DNFs and juntas, respectively:

$$\tilde{O}\left(\frac{\ell^4}{\epsilon^2}\left(\frac{\ell^4}{\epsilon^4} n^{9/2} \frac{\ell^{\Theta(1)}}{\epsilon^{\Theta(1)}}\right)^3\right) \qquad \tilde{O}\left(2^k k \cdot 2^{\theta(k)} n^{9/2} 2^{4(k-1)}\right)$$

A.4. Proof of theorem 15

A statistical query is a real-valued function of a datum $q(x, y) : \mathbb{R}^n \times \{-1, 1\} \to [-1, 1]$. It is issued along with a tolerance τ which is bounded by an inverse polynomial of the problem size. The response is a value r satisfying $\underset{x,y}{\mathbf{E}}(|q(x, y) - r|) \leq \tau$. The algorithm uses the data only to compute empirical gradients $\hat{g}_t$, which are estimates of true gradients g_t. By issuing n queries, one for each dimension, each g_t may be crudely estimated to any inverse polynomial accuracy. A polynomial number of queries thereby replaces direct access to data. (Feldman et al., 2015) describes a more efficient estimation algorithm. Recall the notation of **??**:

$$g_{t,i} = \langle g_t, e_i \rangle = \underset{x,y}{\mathbf{E}_t}\left(\langle x, e_i \rangle y_i\right) = \underset{x,y}{\mathbf{E}}\left(\underbrace{\overbrace{\left(\prod_{j=1}^{t-1}(1 - \left|h_{w_j}(x)\right|)\right) \cdot \langle x, e_i \rangle}^{q(x)} \cdot y}_{q(x,y)}\right)$$

Note the expectation may be written as $\underset{x,y}{\mathbf{E}}(q(x, y)) = \chi(q)$ for a function of just the input. Such correlational queries are theoretically important. In particular, learning from correlational queries is equivalent to evolvability in Valiant's model (Feldman, 2008). This model assumes $\mathsf{opt} = 1$, Furthermore, since the distribution of x is fixed and known, correlational queries may simulate general queries.