Optimal Minimal Margin Maximization with Boosting

Allan Grønlund*† Kasper Green Larsen‡ Alexander Mathiasen§

Abstract

Boosting algorithms produce a classifier by iteratively combining base hypotheses. It has been observed experimentally that the generalization error keeps improving even after achieving zero training error. One popular explanation attributes this to improvements in margins. A common goal in a long line of research, is to maximize the smallest margin using as few base hypotheses as possible, culminating with the AdaBoostV algorithm by [Rätsch and Warmuth, 2005]. The AdaBoostV algorithm was later conjectured to yield an optimal trade-off between number of hypotheses trained and the minimal margin over all training points [Nie et al., 2013]. Our main contribution is a new algorithm refuting this conjecture. Furthermore, we prove a lower bound which implies that our new algorithm is optimal.

1 Introduction

Boosting is one of the most famous and succesful ideas in learning. Boosting algorithms are meta algorithms that produce highly accurate classifiers by combining already existing less accurate classifiers. Probably the most famous boosting algorithm is AdaBoost by Freund and Schapire [Freund and Schapire, 1995], who won the 2003 Gödel Prize for their work.

AdaBoost was designed for binary classification and works by combining base hypotheses learned by a given base learning algorithm into a weighted sum that represents the final classifier. This weighed set of base hypotheses is constructed iteratively in rounds, each round constructing a new base hypothesis that focuses on the training data misclassified by the previous base hypotheses constructed. More precisely, AdaBoost takes training data $D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}\}_{i=1}^n$ and constructs a linear combination classifier $\text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$, where h_t is the base hypothesis learned in the t'th iteration and α_t is the corresponding weight.

It has been proven that AdaBoost decreases the training error exponentially fast if each base hypothesis is slightly better than random guessing on the weighed data set it is trained on [Freund et al., 1999]. Concretely, if ϵ_t is the error of h_t on the weighed data set used to learn h_t then the linear combination has training error at most $\exp(-2\sum_{t=1}^T (1/2 - \epsilon_t)^2)$. If each ϵ_t is at most a half minus a fixed constant, then the training error is less than $1/n$ after $O(\lg n)$ rounds which means the all training points are classified correctly. Quite surprisingly, experiments show that continuing the AdaBoost algorithm even after the training data is perfectly classified, making the model more and more complex, continues to improve generalization [Schapire et al., 1998]. The most prominent approach to explaining this generalization phenomenon considers *margins* [Schapire et al., 1998]. The margin of a point x_i is

$$\text{margin}(x_i) = \frac{y_i \sum_{t=1}^T \alpha_t h_t(x_i)}{\sum_{t=1}^T |\alpha_t|}.$$

For binary classification, if each $h_t(x) \in [-1, +1]$, then the margin of a point is a number between -1 and +1. Notice that a point has positive margin if it is classified correctly and negative margin if it is classified

*All authors contributed evenly.

†Aarhus University. Email: jallan@cs.au.dk.

‡Aarhus University. Email: larsen@cs.au.dk. Supported by a Villum Young Investigator Grant and an AUFF Starting Grant.

§Aarhus University. Email: alexander.mathiasen@gmail.com. Supported by an AUFF Starting Grant.

incorrectly. It has been observed experimentally that the margins of the training points usually increase when training, even after perfectly classifying the training data. This has inspired several bounds on generalization error that depend on the distribution of margins [Schapire et al., 1998, Breiman, 1999, Koltchinskii et al., 2001, Wang et al., 2008, Gao and Zhou, 2013]. The conceptually simplest of these bounds depend only on the minimal margin, which is the margin of the point x_i with minimal margin. The point x_i with minimal margin can be interpreted as the point the classifier struggles the most with. This has inspired a series of algorithms with guarantees on the minimal margin [Breiman, 1999, Grove and Schuurmans, 1998, Bennett et al., 2000, Rätsch and Warmuth, 2002, Rätsch and Warmuth, 2005].

These algorithms have the following goal: Let $\mathcal{H}$ be the (possibly infinite) set of all base hypotheses that may be returned by the base learning algorithm. Suppose the best possible minimal margin on some training data for any linear combination of $h \in \mathcal{H}$ is ρ^*, i.e.

$$\rho^* = \max_{\alpha \neq 0} \left(\min_i \sum_{h \in \mathcal{H}} \frac{y_i \sum_{h \in \mathcal{H}} \alpha_h h(x_i)}{\sum_{h \in \mathcal{H}} |\alpha_h|} \right).$$

Given some precision parameter v, the goal is to construct a linear combination with minimal margin at least $\rho = \rho^* - v$ using as few hypotheses as possible. In this case we say that the linear combination has a gap of v. The current state of the art is AdaBoostV [Rätsch and Warmuth, 2005]. It guarantees a gap of v using $O(\lg(n)/v^2)$ hypotheses. It was later conjectured that there exists data sets D and a corresponding set of base hypotheses $\mathcal{H}$, such that any linear combination of base hypotheses from $\mathcal{H}$ must use at least $\Omega(\lg(n)/v^2)$ hypotheses to achieve a gap of v for any $\sqrt{\lg n/n} \leq v \leq a_1$ for some constant $a_1 > 0$. This would imply optimality of AdaBoostV. This conjecture was published as an open problem in the Journal of Machine Learning Research [Nie et al., 2013].

Our main contribution is a refutal of this conjecture. We refute the conjecture by introducing a new algorithm called SparsiBoost, which guarantees a gap of v with just $T = O(\lg(nv^2)/v^2)$ hypotheses. When $v \leq n^{o(1)}/\sqrt{n}$, SparsiBoost has $T = O(\lg(n^{o(1)})/v^2) = o(\lg(n)/v^2)$, which is asymptotically better than AdaBoostV's $T = O(\lg(n)/v^2)$ guarantee. Moreover, it also refutes the conjectured lower bound. Our algorithm involves a surprising excursion to the field of combinatorial discrepancy minimization. We also show that our algorithm is the best possible. That is, there exists data sets D and corresponding set of base hypotheses $\mathcal{H}$, such that any linear combination of base hypotheses from $\mathcal{H}$ with a gap of v, must use at least $T = \Omega(\lg(nv^2)/v^2)$ hypotheses.

This work thus provides the final answer to over a decade's research into understanding the trade-off between minimal margin and the number of hypotheses: Given a gap v, the optimal number of hypotheses is $T = \Theta(\lg(nv^2)/v^2)$ for any $\sqrt{1/n} \leq v \leq a_1$ where $a_1 > 0$ is a constant. Notice that smaller values for v are irrelevant since it is always possible to achieve a gap of zero using $n + 1$ base hypotheses. This follows from Carathéodory's Theorem.

1.1 Previous Work on Minimal Margin

Upper Bounds. [Breiman, 1999] introduced Arc-GV, which was the first algorithm that guaranteed to find a finite number of hypotheses $T < \infty$ with gap zero ($v = 0$). As pointed out by [Rätsch and Warmuth, 2005], one can think of Arc-GV as a subtle variant of AdaBoost where the weights α_t of the hypotheses are slightly changed. If AdaBoost has hypothesis weight α_t, then Arc-GV chooses the hypothesis weight $\alpha'_t = \alpha_t + x$ for some x that depends on the minimal margin of $h_1, \ldots, h_t$. A few years later, [Grove and Schuurmans, 1998] and [Bennett et al., 2000] introduced DualLPBoost and LPBoost which both have similar guarantees.

[Rätsch and Warmuth, 2002] introduced AdaBoost$_\rho$, which was the first algorithm to give a guarantee on the gap achieved in terms of the number of hypotheses used. Their algorithm takes a parameter $\rho \leq \rho^*$ that serves as the target margin one would like to achieve. It then guarantees a minimal margin of $\rho - \mu$ using $T = O(\lg(n)/\mu^2)$ hypotheses. One would thus like to choose $\rho = \rho^*$. If ρ^* is unknown, it can be found up to an additive approximation of v by binary searching using AdaBoost$_\rho$. This requires an additional $O(\lg 1/v)$ calls to AdaBoost$_\rho$, resulting in $O(\lg(n)/v^2) \lg(1/v)$ iterations of training a base hypothesis to find the desired linear combination of $T = O(\lg(n)/v^2)$ base hypotheses. Similar to Arc-GV, AdaBoost$_\rho$ differs from AdaBoost only in choosing the weights α_t. Instead of having the additional term depend on the minimal margin of $h_1, \ldots, h_t$, it depends only on the estimation parameter ρ.

A few years later, [Rätsch and Warmuth, 2005] introduced AdaBoostV. It is a clever extension of AdaBoost$_\rho$ that uses an adaptive estimate of ρ^* to remove the need to binary search for it. It achieves a gap of v using $T = O(\lg(n)/v^2)$ base hypotheses and no extra iterations of training.

Lower Bounds. [Klein and Young, 1999] showed a lower bound for a seemingly unrelated game theoretic problem. It was later pointed out by [Nie et al., 2013] that their result implies the following lower bound for boosting: there exists a data set of n points and a corresponding set of base hypotheses $\mathcal{H}$, such that any linear combination of $T \in [\lg n; \sqrt{n}]$ base hypotheses must have a gap of $v = \Omega(\sqrt{\lg(n)/T})$. Rewriting in terms of T we get $T = \Omega\left(\lg(n)/v^2\right)$ for $\sqrt{\lg(n)}/n^{1/4} \leq v \leq a_1$ for some constant $a_1 > 0$.

[Nie et al., 2013] conjectured that Klein and Young's lower bound of $v = \Omega(\sqrt{\lg(n)/T})$ holds for all $T \leq a_1 \cdot n$ for some constant $a_1 > 0$. Rewriting in terms of T, they conjecture that $T = \Omega(\lg(n)/v^2)$ holds for $\sqrt{a_2/n} \leq v \leq a_3$ where $a_2, a_3 > 0$ are some constants.

1.2 Our Results On Minimal Margin

Our main result is a novel algorithm, called SparsiBoost, which refutes the conjectured lower bound in [Nie et al., 2013]. Concretely, SparsiBoost guarantees a gap of v with just $T = O(\lg(nv^2)/v^2)$ hypotheses. At a first glance it might seem SparsiBoost violates the lower bound of Klein and Young. Rewriting in terms of v, our upper bound becomes $v = O(\sqrt{\lg(n/T)/T})$ (see Appendix A for details). When $T \leq \sqrt{n}$ (the range of parameters where their lower bound applies), this becomes $v = O(\sqrt{\lg(n)/T})$ which does not violate Klein and Young's lower bound. Moreover, our upper bound explains why both [Klein and Young, 1999] and [Nie et al., 2013] were not able to generalize the lower bound to all $T = O(n)$: When $T = n^{1-o(1)}$, our algorithm achieves a gap of $v = O(\sqrt{\lg(n^{o(1)})/T}) = o(\sqrt{\lg(n)/T})$.

The high level idea of SparsiBoost is as follows: Given a desired gap v, we use AdaBoostV to find $m = O(\lg(n)/v^2)$ hypotheses $h_1, \ldots, h_m$ and weights $w_1, \ldots, w_m$ such that $\sum_i w_i h_i$ achieves a gap of $v/2$. We then carefully "sparsify" the vector $w = (w_1, \ldots, w_m)$ to obtain another vector w' that has at most $T = O(\lg(nv^2)/v^2)$ non-zeroes. Our sparsification is done such that the margin of every single data point changes by at most $v/2$ when replacing $\sum_i w_i h_i$ by $\sum_i w'_i h_i$. In particular, this implies that the minimum margin, and hence gap, changes by at most $v/2$. We can now safely ignore all hypotheses h_i where $w'_i = 0$ and we have obtained the claimed gap of at most $v/2 + v/2 = v$ using $T = O(\lg(nv^2)/v^2)$ hypotheses.

Our algorithm for sparsifying w gives a general method for sparsifying a vector while approximately preserving a matrix-vector product. We believe this result may be of independent interest and describe it in more detail here: The algorithm is given as input a matrix $U \in [-1, +1]^{n \times m}$ and a vector $w \in \mathbb{R}^m$ where $\|w\|_1 = 1$. It then finds a vector w' such that $\|Uw - Uw'\|_\infty = O(\lg(n/T)/T)$, $\|w'\|_0 \leq T$ and $\|w'\|_1 = 1$. Here $\|x\|_1 = \sum_i |x_i|$, $\|x\|_\infty = \max_i |x_i|$ and $\|x\|_0$ denotes the number of non-zero entries of x. When we use this result in SparsiBoost, we will define the matrix U as the "margin matrix" that has $u_{ij} = y_i h_j(x_i)$. Then $(Uw)_i = \text{margin}(x_i)$ and the guarantee $\|Uw - Uw'\|_\infty = O(\lg(n/T)/T)$ will ensure that the margin of every single point changes by at most $O(\lg(n/T)/T)$ if we replace the weights w by w'. Our algorithm for finding w' is based on a novel connection to the celebrated but seemingly unrelated "six standard devitations suffice" result by [Spencer, 1985] from the field of combinatorial discrepancy minimization.

When used in SparsiBoost, the matrix U is defined from the output of AdaBoostV, but the vector sparsification algorithm could just as well be applied to the hypotheses output by any boosting algorithm. Thus our results give a general method for sparsifying a boosting classifier while approximately preserving the margins of all points.

We complement our new upper bound with a matching lower bound. More concretely, we prove that there exists data sets D of n points and a corresponding set of base hypotheses $\mathcal{H}$, such that any linear combination of T base hypotheses must have a gap of at least $v = \Omega(\sqrt{\lg(n/T)/T})$ for any $\lg n \leq T \leq a_1 n$ where $a_1 > 0$ is a constant. Rewriting in terms of T, one must use $T = \Omega(\lg(nv^2)/v^2)$ hypotheses from $\mathcal{H}$ to achieve a gap of v. This holds for any v satisfying $\sqrt{a_2/n} < v \leq a_3$ for constants $a_2, a_3 > 0$ (see Appendix A for details).

Interestingly, our lower bound proof also uses the discrepancy minimization upper bound by [Spencer, 1985] in a highly non-trivial way. Our lower bound also shows that our vector sparsification algorithm is optimal for any $T \leq n/C$ for some universal constant $C > 0$.

1.3 Doubts on Margin Theory and other Margin Bounds

The first margin bound on boosted classifiers was introduced by [Schapire et al., 1998]. Shortly after, [Breiman, 1999] introduced a sharper minimal margin bound alongside Arc-GV. Experimentally Breimann found that Arc-GV produced better margins than AdaBoost on 98 percent of the training data, however, AdaBoost still obtained a better test error. This seemed to contradict margin theory: according to margin theory, better margins should imply better generalization. This caused Breimann to doubt margin theory. It was later discovered by [Reyzin and Schapire, 2006] that the comparison was unfair due to a difference in the complexity of the base hypotheses used by AdaBoost and Arc-GV. [Reyzin and Schapire, 2006] performed a variant of Breimann's experiments with decision stumps to control the hypotheses complexity. They found that even though Arc-GV produced a better minimal margin, AdaBoost produced a larger margin on almost all other points [Reyzin and Schapire, 2006] and that AdaBoost generalized better.

A few years later, [Wang et al., 2008] introduced a sharper margin bound than Breimann's minimal margin bound. The generalization bound depends on a term called the *Equilibrium Margin*, which itself depends on the margin distribution in a highly non-trivial way. This was followed by the k-margin bound by [Gao and Zhou, 2013] that provide generalization bounds based on the k'th smallest margin for any k. The bound gets weaker with increasing k, but stronger with increasing margin. In essence, this means that we get stronger generalization bounds if the margins are large for small values of k.

Recall from the discussion in Section 1.2 that our sparsification algorithm preserves all margins to within $O(\sqrt{\lg(n/T)/T})$ additive error. We combined this result with AdaBoostV to get our algorithm SparsiBoost which obtained an optimal trade-off between minimal margin and number of hypotheses. While minimal margin might be insufficient for predicting generalization performance, our sparsification algorithm actually preserves the full distribution of margins. Thus according to margin theory, the sparsified classifier should approximately preserve the generalization performance of the full unsparsified classifier. To demonstrate this experimentally, we sparsified a classifier trained with LightGBM [Ke et al., 2017], a highly efficient open-source implementation of Gradient Boosting [Mason et al., 2000, Friedman, 2001]. We compared the margin distribution and the test error of the sparsified classifier against a LightGBM classifier trained directly to have the same number of hypotheses. Our results (see Section 4) show that the sparsified classifier has a better margin distribution and indeed generalize better than the standard LightGBM classifier.

2 SparsiBoost

In this section we introduce SparsiBoost. The algorithm takes the following inputs: training data D, a target number of hypotheses T and a base learning algorithm A that returns hypotheses from a class $\mathcal{H}$ of possible base hypotheses. SparsiBoost initially trains $c \cdot T$ hypotheses for some appropriate c, by running AdaBoostV with the base learning algorithm A. It then removes the extra $c \cdot T - T$ hypotheses while attempting to preserve the margins on all training examples.

In more detail, let $h_1, ..., h_{cT} \in \mathcal{H}$ be the hypotheses returned by AdaBoostV with weights $w_1, ..., w_{cT}$. Construct a margin matrix U that contains the margin of every hypothesis h_j on every point x_i such that $u_{ij} = y_i h_j(x_i)$. Let w be the vector of hypothesis weights, meaning that the j'th coordinate of w has the weight w_j of hypothesis h_j. Normalize $w = w/\|w\|_1$ such that $\|w\|_1 = 1$. The product Uw is then a vector that contains the margins of the final linear combination on all points: $(Uw)_i = y_i \sum_{j=1}^{cT} w_j h_j(x_i) = \text{margin}(x_i)$. Removing hypotheses while preserving the margins can be formulated as sparsifying w to w' while minimizing $\|Uw - Uw'\|_\infty$ subject to $\|w'\|_0 \leq T$ and $\|w'\|_1 = 1$.

See Algorithm 1 for pseudocode.

We still haven't described how to find w' with the guarantees shown in Algorithm 1 step 4., i.e. a w' with $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2 + n/T)/T})$. It is not even clear that such w' exists, much less so that it can be

Algorithm 1 SparsiBoost

Input: Training data $D = \{(x_i, y_i)\}_{i=1}^n$ where $x_i \in X$ for some input space X and $y_i \in \{-1, +1\}$. Target number of hypotheses T and base learning algorithm A.

Output: Hypotheses $h_1, \ldots, h_k$ and weights $w_1, \ldots, w_k$ with $k \le T$, such that $\sum_i w_i h_i$ has gap $O(\sqrt{\lg(2 + n/T)/T})$ on D.

1. Run AdaBoostV with base learning algorithm A on training data D to get cT hypotheses $h_1, \ldots, h_{cT}$ and weights $w_1, \ldots, w_{cT}$ for the integer $c = \lceil \lg(n)/\lg(2 + n/T) \rceil$.
2. Construct margin matrix $U \in [-1, +1]^{n \times cT}$ where $u_{ij} = y_i h_j(x_i)$.
3. Form the vector w with i'th coordinate w_i and normalize $w \leftarrow w/\|w\|_1$ so $\|w\|_1 = 1$.
4. Find w' such that $\|w'\|_0 \le T$, $\|w'\|_1 = 1$ and $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2 + n/T)/T})$.
5. Let $\pi(j)$ denote the index of the j'th non-zero entry of w'.
6. **Return** hypotheses $h_{\pi(1)}, \ldots, h_{\pi(\|w'\|_0)}$ with weights $w'_{\pi(1)}, \ldots, w'_{\pi(\|w'\|_0)}$.

found efficiently. Before we dive into the details of how to find w', we briefly demonstrate that indeed such a w' would be sufficient to establish our main theorem:

Theorem 2.1. *SparsiBoost is guaranteed to find a linear combination w' of at most T base hypotheses with gap $v = O(\sqrt{\lg(2 + n/T)/T})$.*

Proof. We assume throughout the proof that a w' with the guarantees claimed in Algorithm 1 can be found. Suppose we run AdaBoostV to get cT base hypotheses $h_1, ..., h_{cT}$ with weights $w_1, ..., w_{cT}$. Let ρ_{cT} be the minimal margin of the linear combination $\sum_i w_i h_i$ on the training data D, and let ρ^* be the optimal minimal margin over all linear combinations of base hypotheses from $\mathcal{H}$. As proved in [Rätsch and Warmuth, 2005], AdaBoostV guarantees that the gap is bounded by $\rho^* - \rho_{cT} = O(\sqrt{\lg(n)/(cT)})$. Normalize $w = w/\|w\|_1$ and let U be the margin matrix $u_{ij} = y_i h_j(x_i)$ as in Algorithm 1. Then $\rho_{cT} = \min_i (Uw)_i$. From our assumption, we can efficiently find w' such that $\|w'\|_1 = 1$, $\|w'\|_0 \le T$ and $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2 + n/T)/T})$. Consider the hypotheses that correspond to the non-zero entries of w'. There are at most T. Let ρ_T be their minimal margin when using the corresponding weights from w'. Since w' has unit ℓ_1-norm, it follows that $\rho_T = \min_i (Uw')_i$ and thus $|\rho_T - \rho_{cT}| \le \max_i |(Uw)_i - (Uw')_i|$, i.e. $|\rho_{cT} - \rho_T| \le \|Uw - Uw'\|_\infty = O(\sqrt{\lg(2 + n/T)/T})$. We therefore have:

$$\rho^* - \rho_T = (\rho^* - \rho_{cT}) + (\rho_{cT} - \rho_T) \le$$
$$O(\sqrt{\lg(n)/(cT)}) + O(\sqrt{\lg(2 + n/T)/T}).$$

By choosing $c = \lg(n)/\lg(2 + n/T)$ (as in Algorithm 1) we get that $\rho^* - \rho_T = O(\sqrt{\lg(2 + n/T)/T})$. $\square$

The core difficulty in our algorithm is thus finding an appropriate w' (step 4 in Algorithm 1) and this is the focus of the remainder of this section. Our algorithm for finding w' gives a general method for sparsifying a vector w while approximately preserving every coordinate of the matrix-vector product Uw for some input matrix U. The guarantees we give are stated in the following theorem:

Theorem 2.2. *(Sparsification Theorem) For all matrices $U \in [-1, +1]^{n \times m}$, all $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$ and all $T \le m$, there exists a vector w' where $\|w'\|_1 = 1$ and $\|w'\|_0 \le T$, such that $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2 + n/T)/T})$.*

Theorem 2.2 is exactly what was needed in the proof of Theorem 2.1. Our proof of Theorem 2.2 will be constructive in that it gives an algorithm for finding w'. To keep the proof simple, we will argue about running time at the end of the section.

The first idea in our algorithm and proof of Theorem 2.2, is to reduce the problem to a simpler task, where instead of reducing the number of hypotheses directly to T, we only halve the number of hypotheses:

Lemma 2.1. *For all matrices $U \in [-1,+1]^{n\times m}$ and $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$, there exists w' where $\|w'\|_0 \le \|w\|_0/2$ and $\|w'\|_1 = 1$, such that $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$.*

To prove Theorem 2.2 from Lemma 2.1, we can repeatedly apply Lemma 2.1 until we are left with a vector with at most T non-zeroes. Since the loss $O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$ has a $\sqrt{1/\|w\|_0}$ factor, we can use the triangle inequality to conclude that the total loss is a geometric sum that is asymptotically dominated by the very last invocation of the halving procedure. Since the last invocation has $\|w\|_0 > T$ (otherwise we would have stopped earlier), we get a total loss of $O(\sqrt{\lg(2+n/T)/T})$ as desired. The formal proof can be found in Section 2.1.

The key idea in implementing the halving procedure Lemma 2.1 is as follows: Let $\pi(j)$ denote the index of the j'th non-zero in w and let $\pi^{-1}(j)$ denote the index i such that w_i is the j'th non-zero entry of w. First we construct a matrix A where the j'th column of A is equal to the $\pi(j)$'th column of U scaled by the weight $w_{\pi(j)}$. The sum of the entries in the i'th row of A is then equal to the i'th entry of Uw (since $\sum_j a_{ij} = \sum_j w_{\pi^{-1}(j)} u_{i\pi^{-1}(j)} = \sum_{j:w_j\neq 0} w_j u_{ij} = \sum_j w_j u_{ij} = (Uw)_i$). Minimizing $\|Uw - Uw'\|_\infty$ can then be done by finding a subset of columns of A that approximately preserves the row sums. This is formally expressed in Lemma 2.2 below. For ease of notation we define $a \pm [x]$ to be the interval $[a-x, a+x]$ an $\pm[x]$ to be the interval $[-x, x]$.

Lemma 2.2. *For all matrices $A \in [-1,1]^{n\times T}$ there exists a submatrix $\hat{A} \in [-1,1]^{n\times k}$ consisting of $k \le T/2$ distinct columns from A, such that for all i, it holds that $\sum_{j=1}^{k} \hat{a}_{ij} \in \frac{1}{2}\sum_{j=1}^{T} a_{ij} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]$.*

Intuitively we can now use Lemma 2.2 to select a subset S of at most $T/2$ columns in A. We can then replace the vector w with w' such that $w'_i = 2w_i$ if $i = \pi^{-1}(j)$ for some $j \in S$ and $w'_i = 0$ otherwise. In this way, the i'th coordinate $(Uw')_i$ equals the i'th row sum in $\hat{A}$, scaled by a factor two. By Lemma 2.2, this in turn approximates the i'th row sum in A (and thus $(Uw)_i$) up to additively $O(\sqrt{T\lg(2+n/T)})$.

Unfortunately our procedure is not quite that straightforward since $O(\sqrt{T\lg(2+n/T)})$ is way too large compared to $O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0}) = O(\sqrt{\lg(2+n/T)/T})$. Fortunately Lemma 2.2 only needs the coordinates of A to be in $[-1,1]$. We can thus scale A by $1/\max_i |w_i|$ and still satisfy the constraints. This in turn means that the loss is scaled down by a factor $\max_i |w_i|$. However, $\max_i |w_i|$ may be as large as 1 for highly unbalanced vectors. Therefore, we start by copying the largest $T/3$ entries of w to w' and invoke Lemma 2.2 twice on the remaining $2T/3$ entries. This ensures that the $O(\sqrt{T\lg(2+n/T)})$ loss in Lemma 2.2 gets scaled by a factor at most $3/T$ (since $\|w\|_1 = 1$, all remaining coordinates are less than or equal to $3/T$), while leaving us with at most $T/3 + (2T/3)/4 = T/3 + T/6 = T/2$ non-zero entries as required. Since we normalize by at most $3/T$, the error becomes $\|Uw - Uw'\|_\infty = O(\sqrt{T\lg(2+n/T)}/T) = O(\sqrt{\lg(2+n/T)/T})$ as desired. As a last technical detail, we also need to ensure that w' satisfies $\|w'\|_1 = 1$. We do this by adding an extra row to A such that $a_{(n+1)j} = w_j$. In this way, preserving the last row sum also (roughly) preserves the ℓ_1-norm of w and we can safely normalize w' as $w' \leftarrow w'/\|w\|_1$. The formal proof is given in Section 2.2.

The final step of our algorithm is thus to select a subset of at most half of the columns from a matrix $A \in [-1,1]^{n\times T}$, while approximately preserving all row sums. Our idea for doing so builds on the following seminal result by Spencer:

Theorem 2.3. *(Spencer's Theorem [Spencer, 1985]) For all matrices $A \in [-1,+1]^{n\times T}$ with $T \le n$, there exists $x \in \{-1,+1\}^T$ such that $\|Ax\|_\infty = O(\sqrt{T\ln(en/T)})$. For all matrices $A \in [-1,+1]^{n\times T}$ with $T > n$, there exists $x \in \{-1,+1\}^T$ such that $\|Ax\|_\infty = O(\sqrt{n})$.*

We use Spencer's Theorem as follows: We find a vector $x \in \{-1,+1\}^T$ with $\|Ax\|_\infty = O(\sqrt{T\ln(en/T)})$ if $T \le n$ and with $\|Ax\|_\infty = O(\sqrt{n}) = O(\sqrt{T})$ if $T > n$. Thus we always have $\|Ax\|_\infty = O(\sqrt{T\lg(2+n/T)})$. Consider now the i'th row of A and notice that $|\sum_{j:x_j=1} a_{ij} - \sum_{j:x_j=-1} a_{ij}| \le \|Ax\|_\infty$. That is, for every single row, the sum of the entries corresponding to columns where x is 1, is almost equal (up to $\pm\|Ax\|_\infty$) to the sum over the columns where x is -1. Since the two together sum to the full row sum, it follows that the subset of columns with $x_i = 1$ *and* the subset of columns with $x_i = -1$ *both* preserve the row sum as required by Lemma 2.2. Since x has at most $T/2$ of either $+1$ or -1, it follows that we can find the desired subset of columns. We give the formal proof of Lemma 2.2 using Spencer's Theorem in Section 2.3

Algorithm 2 Sparsification

Input: Matrix $U \in [-1,1]^{n\times m}$, vector $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$ and target $T \le m$.
Output: A vector $w' \in \mathbb{R}^m$ with $\|w'\|_1 = 1, \|w'\|_0 \le T$ and $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/T)/T})$.

1. Let $w' \leftarrow w$.
2. **While** $\|w'\|_0 > T$:
3. Let R be the indices of the $\|w'\|_0/3$ entries in w' with largest absolute value.
4. Let $\omega := \max_{i\notin R} |w_i|$ be the largest value of an entry outside R.
5. **Do Twice**:
6. Let $\pi(1), \pi(2), \dots, ..., \pi(k)$ be the indices of the non-zero entries in w' that are not in R.
7. Let $A \in [-1,1]^{(n+1)\times k}$ have $a_{ij} = u_{i\pi(j)} w'_{\pi(j)}/\omega$ for $i \le n$ and $a_{(n+1)j} = |w'_{\pi(j)}|/\omega$.
8. Invoke Spencer's Theorem to find $x \in \{-1,1\}^k$ such that $\|Ax\|_\infty = O(\sqrt{k \lg(2+n/k)})$.
9. Let $\sigma \in \{-1,1\}$ denote the sign such that $x_i = \sigma$ for at most $k/2$ indices i.
10. Update w'_i as follows:
11. If there is a j such that $i = \pi^{-1}(j)$ and $x_j = \sigma$: set $w'_i \leftarrow 2w'_i$.
12. If there is a j such that $i = \pi^{-1}(j)$ and $x_j \neq \sigma$: set $w'_i \leftarrow 0$.
13. Otherwise ($i \in R$ or $w'_i = 0$): set $w'_i \leftarrow w'_i$.
14. Update $w' \leftarrow w'/\|w'\|_1$.
15. **Return** w'.

We have summarized the entire sparsification algorithm in Algorithm 2.

We make a few remarks about our sparsification algorithm and SparsiBoost.

Running Time. While Spencer's original result (Theorem 2.3) is purely existential, recent follow up work [Lovett and Meka, 2015] show how to find the vector $x \in \{-1,+1\}^n$ in expected $\tilde{O}((n+T)^3)$ time, where $\tilde{O}$ hides polylogarithmic factors. A small modification to the algorithm was suggested in [Larsen, 2017]. This modification reduces the running time of Lovett and Meka's algorithm to expected $\tilde{O}(nT + T^3)$. This is far more desirable as T tends to be much smaller than n in boosting. Moreover, the nT term is already paid by simply running AdaBoostV. Using this in Step 7. of Algorithm 2, we get a total expected running time of $\tilde{O}(nT + T^3)$. We remark that these algorithms are randomized and lead to different vectors x on different executions.

Non-Negativity. Examining Algorithm 2, we observe that the weights of the input vector are only ever copied, set to zero, or scaled by a factor two. Hence if the input vector w has non-negative entries, then so has the final output vector w'. This may be quite important if one interprets the linear combination over hypotheses as a probability distribution.

Importance Sampling. Another natural approach one might attempt in order to prove our sparsification result, Theorem 2.2, would be to apply importance sampling. Importance sampling samples T entries from w with replacement, such that each entry i is sampled with probability $|w_i|$. It then returns the vector w' where coordinate i is equal to $\text{sign}(w_i)n_i/T$ where n_i denotes the number of times i was sampled and $\text{sign}(w_i) \in \{-1,1\}$ gives the sign of w_i. Analysing this method gives a w' with $\|Uw - Uw'\|_\infty = \Theta(\sqrt{\lg(n)/T})$ (with high probability), i.e. slightly worse than our approach based on discrepancy minimization. The loss in the lg is enough that if we use importance sampling in SparsiBoost, then we get no improvement over simply stopping AdaBoostV after T iterations.

2.1 Repeated Halving

As discussed earlier, our matrix-vector sparsification algorithm (Theorem 2.2 and Algorithm 2) was based on repeatedly invoking Lemma 2.1, which halves the number of non-zero entries. In this section, we prove formally that the errors resulting from these halving steps are dominated by the very last round of halving. For convenience, we first restate the two results:

Restatement of Theorem 2.2. *(Sparsification Theorem) For all matrices $U \in [-1,+1]^{n\times m}$, all $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$ and all $T \le m$, there exists a vector w' where $\|w'\|_1 = 1$ and $\|w'\|_0 \le T$, such that $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/T)/T})$.*

Restatement of Lemma 2.1. *For all matrices $U \in [-1,+1]^{n\times m}$ and $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$, there exists w' where $\|w'\|_0 \le \|w\|_0/2$ and $\|w'\|_1 = 1$, such that $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$.*

We thus set out to prove Theorem 2.2 assuming Lemma 2.1.

Proof. Let us call the initial weight vector $w^{(0)} = w$. We use Lemma 2.1 repeatedly and get $w^{(1)}, w^{(2)}, ..., w^{(k)} = w'$ such that $\|w'\|_0 \le T$ as wanted and every $w^{(i)}$ has $\|w^{(i)}\|_1 = 1$. Let $T_i = \|w^{(i)}\|_0$ and notice this gives a sequence of numbers $T_0, T_1, ..., T_k$ where $T_i \le T_{i-1}/2$, $T_{k-1} > T$ and $T_k \le T$. In particular, it holds that $T_{k-1}2^{k-i-1} \le T_i$. The total difference $\|Uw^{(0)} - Uw^{(k)}\|_\infty$ is then by the triangle inequality no more than $\sum_{i=0}^{k-1} \|Uw^{(i)} - Uw^{(i+1)}\|_\infty$. Each of these terms are bounded by Lemma 2.1 which gives us

$$O\left(\sum_{i=0}^{k-1} \sqrt{\lg(2+n/T_i)/T_i}\right) = O\left(\sum_{i=0}^{k-1} \sqrt{\lg(2+n/T_{k-1})/(T_{k-1}2^{k-i-1})}\right) =$$

$$O\left(\sum_{i=0}^{\infty} \sqrt{\lg(2+n/T_{k-1})/(T_{k-1}2^{i})}\right) = O\left(\left(\sqrt{\lg(2+n/T_{k-1})/T_{k-1}}\right)\sum_{i=0}^{\infty} 1/\sqrt{2^i}\right) =$$

$$O\left(\sqrt{\lg(2+n/T)/T}\right).$$

The last step follows since $T_{k-1} > T$. We have thus shown that the final vector $w' = w^{(k)}$ has $\|w'\|_0 \le T$, $\|w'\|_1 = 1$ and $\|Uw - Uw'\|_\infty = \|Uw^{(0)} - Uw^{(k)}\|_\infty = O\left(\sqrt{\lg(2+n/T)/T}\right)$. This completes the proof of Theorem 2.2. □

2.2 Halving via Row-Sum Preservation

In this section, we give the formal details of how to use our result on row-sum preservation to halve the number of non-zeroes in a vector w. Let us recall the two results:

Restatement of Lemma 2.1. *For all matrices $U \in [-1,+1]^{n\times m}$ and $w \in \mathbb{R}^m$ with $\|w\|_1 = 1$, there exists w' where $\|w'\|_0 \le \|w\|_0/2$ and $\|w'\|_1 = 1$, such that $\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$.*

Restatement of Lemma 2.2. *For all matrices $A \in [-1,1]^{n\times T}$ there exists a submatrix $\hat{A} \in [-1,1]^{n\times k}$ consisting of $k \le T/2$ distinct columns from A, such that for all i, it holds that $\sum_{j=1}^{k} \hat{a}_{ij} \in \frac{1}{2}\sum_{j=1}^{T} a_{ij} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]$.*

We now use Lemma 2.2 to prove Lemma 2.1.

Proof. Our procedure corresponds to steps 2.-13. in Algorithm 2. To clarify the proof, we expand the steps a bit and introduce some additional notation. Let w be the input vector and let R be the indices of the $\|w\|_0/3$ entries in w with largest absolute value. Define $\bar{w}$ such that $\bar{w}_i = w_i$ if $i \in R$ and $\bar{w}_i = 0$ otherwise, that is, $\bar{w}$ contains the largest $\|w\|_0/3$ entries of w and is zero elsewhere. Similarly, define $\hat{w} = w - \bar{w}$ as the vector containing all but the largest $\|w\|_0/3$ entries of w.

We will use Lemma 2.2 twice in order to obtain a vector w'' with $\|w''\|_0 \leq \|\hat{w}\|_0/4$, $\|w''\|_1 \in \|\hat{w}\|_1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})]$ and $\|U\hat{w} - Uw''\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$. Moreover, w'' will only be non-zero in entries where $\hat{w}$ is also non-zero. We finally set $w' = (\bar{w}+w'')/\|\bar{w}+w''\|_1$ as our sparsified vector.

We first argue that if we can indeed produce the claimed w'', then w' satisfies the claims in Lemma 2.1: Observe that $\|w'\|_0 = \|\bar{w}\|_0 + \|w''\|_0 \leq \|w\|_0/3 + (2\|w\|_0/3)/4 \leq \|w\|_0/2$ as desired. Clearly we also have $\|w'\|_1 = 1$ because of the normalization. Now observe that:

$$\begin{aligned}
\|Uw - Uw'\|_\infty &\leq \|Uw - U(\bar{w}+w'')\|_\infty + \|U(\bar{w}+w'') - Uw'\|_\infty \\
&= \|U(\bar{w}+\hat{w}) - U(\bar{w}+w'')\|_\infty + \|U(w'\|\bar{w}+w''\|_1) - Uw'\|_\infty \\
&= \|U\hat{w} - Uw''\|_\infty + \|Uw'(\|\bar{w}+w''\|_1 - 1)\|_\infty \\
&= O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0}) + \|Uw'\|_\infty(\|\bar{w}+w''\|_1 - 1) \\
&= O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0}) + \|Uw'\|_\infty(\|\bar{w}\|_1 + \|w''\|_1 - 1).
\end{aligned}$$

In the last step, we used that w'' has non-zeroes only where $\hat{w}$ has non-zeroes (and thus the non-zeroes of $\bar{w}$ and w'' are disjoint). Since $\|w'\|_1 = 1$ and all entries of U are in $[-1,1]$, we get $\|Uw'\|_\infty \leq 1$. We also see that:

$$\begin{aligned}
\|\bar{w}\|_1 + \|w''\|_1 - 1 &\in \|\bar{w}\|_1 + \|\hat{w}\|_1 - 1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})] \\
&= \|\bar{w}+\hat{w}\|_1 - 1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})] \\
&= \|w\|_1 - 1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})] \\
&= 0 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})].
\end{aligned}$$

We have thus shown that

$$\|Uw - Uw'\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$$

as claimed. So what remains is to argue that we can find w'' with the claimed properties. Finding w'' corresponds to the Do Twice part of Algorithm 2 (steps 3.-12.). We first compute $\omega = \max_i |\hat{w}_i|$ and let $w'' = \hat{w}$. We then execute the following twice:

1. Define $\pi(1), \ldots, \pi(\ell)$ as the list of all the indices i where $w''_i \neq 0$. Also define $\pi^{-1}(j)$ as the index i such that $i = \pi(j)$, i.e. $\pi^{-1}(j)$ is the index of the j'th non-zero coordinate of w''.

2. Form the matrix $A \in [-1,1]^{(n+1)\times \ell}$ where $a_{ij} = u_{i\pi(j)} w''_{\pi(j)}/\omega$ for $i \leq n$ and $a_{(n+1)j} = |w''_{\pi(j)}|/\omega$.

3. Invoke Lemma 2.2 to obtain a matrix $\hat{A}$ consisting of no more than $k \leq \ell/2$ distinct columns from A where for all rows i, we have $\sum_{j=1}^{k} \hat{a}_{ij} \in \frac{1}{2}\sum_{j=1}^{\ell} a_{ij} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]$.

4. Update w'' as follows:

   (a) We let $w''_i \leftarrow 2w''_i$ if there is a j such that $i = \pi^{-1}(j)$ and j is a column in $\hat{A}$.

   (b) Otherwise we let $w''_i \leftarrow 0$.

After the two executions, we get that the number of non-zeroes in w'' is at most $\|\hat{w}\|_0/4$ as claimed. Step 4. above effectively scales every coordinate of w'' that corresponds to a column that was included in $\hat{A}$ by a factor two. It sets coordinates not chosen by $\hat{A}$ to 0. What remains is to argue that $\|U\hat{w} - Uw''\|_\infty = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})$ and that $\|w''\|_1 \in \|\hat{w}\|_1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})]$. For this, let $A \in [-1,1]^{(n+1)\times T}$ denote the matrix formed in step 2. during the first iteration. Then $T = \|\hat{w}\|_0$. Let $\bar{A} \in [-1,1]^{(n+1)\times k}$ denote the matrix returned in step 3. of the first iteration. Similarly, let $\hat{A}$ denote the matrix formed in step 2. of the second iteration and let $\hat{\hat{A}} \in [-1,1]^{(n+1)\times k'}$ denote the matrix returned in step 3. of the second iteration. We see that $\hat{A} = 2\bar{A}$. By Lemma 2.2, it holds for all rows i that:

$$\begin{aligned} 2\sum_{j=1}^{k'} \hat{\hat{a}}_{ij} &\in \sum_{j=1}^{k} \hat{a}_{ij} \pm \left[O(\sqrt{k\lg(2+n/k)})\right] \\ &= 2\sum_{j=1}^{k} \bar{a}_{ij} \pm \left[O(\sqrt{k\lg(2+n/k)})\right] \\ &\subseteq \left(\sum_{j=1}^{T} a_{ij} \pm \left[O(\sqrt{k\lg(2+n/k)})\right]\right) \pm \left[O(\sqrt{T\lg(2+n/T)})\right] \\ &\subseteq \sum_{j=1}^{T} a_{ij} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]. \end{aligned}$$

But $2\sum_{j=1}^{k'} \hat{\hat{a}}_{ij} = \sum_j u_{ij} w''_j/\omega = (Uw'')_i/\omega$ and $\sum_{j=1}^{T} a_{ij} = \sum_j u_{ij}\hat{w}_j/\omega = (U\hat{w})_i/\omega$. Hence:

$$\begin{aligned} (Uw'')_i/\omega &\in (U\hat{w})_i/\omega \pm \left[O(\sqrt{T\lg(2+n/T)})\right] \Rightarrow \\ (Uw'')_i &\in (U\hat{w})_i \pm \left[O(\omega\sqrt{T\lg(2+n/T)})\right] \end{aligned}$$

which implies that $\|U\hat{w} - Uw''\|_\infty = O(\omega\sqrt{T\lg(2+n/T)})$. But $T = \|\hat{w}\|_0 \le \|w\|_0$ and $\omega = \max_i |\hat{w}|_i$. Since $\|w\|_1 = 1$ and $\bar{w}$ contained the largest $\|w\|_0/3$ entries, we must have $\omega \le 3/\|w\|_0$. Inserting this, we conclude that

$$\|U\hat{w} - Uw''\|_\infty = O(\sqrt{\|w\|_0\lg(2+n/\|w\|_0)}/\|w\|_0) = O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0}).$$

The last step is to prove that $\|w''\|_1 \in \|\hat{w}\|_1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})]$. Here we focus on row $(n+1)$ of A and use that we showed above that $2\sum_{j=1}^{k'} \hat{\hat{a}}_{(n+1)j} = \sum_{j=1}^{T} a_{(n+1)j} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]$. This time, we have $\sum_{j=1}^{T} a_{(n+1)j} = \sum_{j=1}^{T} |\hat{w}_j|/\omega = \|\hat{w}\|_1/\omega$ and $2\sum_{j=1}^{k'} \hat{\hat{a}}_{(n+1)j} = \sum_j |w''_j|/\omega = \|w''\|_1/\omega$. Therefore we conclude that $\|w''\|_1 \in \|\hat{w}\|_1 \pm \left[O(\omega\sqrt{T\lg(2+n/T)})\right] \subseteq \|\hat{w}\|_1 \pm [O(\sqrt{\lg(2+n/\|w\|_0)/\|w\|_0})]$ as claimed. □

2.3 Finding a Column Subset

In this section we give the detailed proof of how to use Spencer's theorem to select a subset of columns in a matrix while approximately preserving its row sums. The two relevant lemmas are restated here for convenience:

Restatement of Lemma 2.2. *For all matrices $A \in [-1,1]^{n\times T}$ there exists a submatrix $\hat{A} \in [-1,1]^{n\times k}$ consisting of $k \le T/2$ distinct columns from A, such that for all i, it holds that $\sum_{j=1}^{k} \hat{a}_{ij} \in \frac{1}{2}\sum_{j=1}^{T} a_{ij} \pm \left[O(\sqrt{T\lg(2+n/T)})\right]$.*

and

Restatement of Theorem 2.3. *(Spencer's Theorem [Spencer, 1985]) For all matrices $A \in [-1,+1]^{n\times T}$ with $T \le n$, there exists $x \in \{-1,+1\}^T$ such that $\|Ax\|_\infty = O(\sqrt{T\ln(en/T)})$. For all matrices $A \in [-1,+1]^{n\times T}$ with $T > n$, there exists $x \in \{-1,+1\}^T$ such that $\|Ax\|_\infty = O(\sqrt{n})$.*

We now prove Lemma 2.2 using Spencer's Theorem:

Proof. Let $A \in [-1,1]^{n\times T}$ as in Lemma 2.2 and let a_i denote the i'th row of A. By Spencer's Theorem there must exist $x \in \{-1,+1\}^T$ s.t. $\|Ax\|_\infty = \max_i |\langle a_i, x\rangle| = O(\sqrt{T\ln(en/T)})$ if $T \le n$ and $\|Ax\|_\infty = O(\sqrt{n}) = O(\sqrt{T})$ if $T > n$. This ensures that $\|Ax\|_\infty = O(\sqrt{T\lg(2+n/T)})$ Either the number of $+1$'s or -1's in x will be $\le \frac{T}{2}$. Assume without loss of generality that the number of $+1$'s is $k \le \frac{T}{2}$. Construct a matrix $\hat{A}$ with columns of A corresponding to the entries of x that are $+1$. Then $\hat{A} \in [-1,1]^{n\times k}$ for $k \le \frac{T}{2}$ as wanted. It is then left to show that $\hat{A}$ preserves the row sums. For all i, we have:

$$
\begin{aligned}
|\langle a_i, x\rangle| = \left|\sum_{j=1}^{T} a_{ij}x_j\right| = O(\sqrt{T\lg(2+n/T)}) \quad &\Rightarrow \quad \sum_j a_{ij}x_j \in \pm\left[O(\sqrt{T\lg(2+n/T)})\right] \\
&\Rightarrow \sum_{j:x_j=+1} a_{ij} - \sum_{j:x_j=-1} a_{ij} \in \pm\left[O(\sqrt{T\lg(n/T)})\right] \\
&\Rightarrow \sum_{j:x_j=+1} a_{ij} \in \sum_{j:x_j:-1} a_{ij} \pm\left[O(\sqrt{T\lg(n/T)})\right].
\end{aligned}
$$

Using this we can bound the i'th row sum of $\hat{A}$

$$
\begin{aligned}
\sum_{j=1}^{k} \hat{a}_{ij} = \sum_{j:x_j=+1} a_{ij} &= \frac{1}{2}\sum_{j:x_j=+1} a_{ij} + \frac{1}{2}\sum_{j:x_j=+1} a_{ij} \\
&\in \frac{1}{2}\sum_{j:x_j=+1} a_{ij} + \frac{1}{2}\left(\sum_{j:x_j=-1} a_{ij} \pm\left[O(\sqrt{T\lg(2+n/T)})\right]\right) \\
&= \frac{1}{2}\sum_j a_{ij} \pm\left[O(\sqrt{T\lg(2+n/T)})\right].
\end{aligned}
$$

This concludes the proof. □

3 Lower Bound

In this section, we prove our lower bound stating that there exist a data set and corresponding set of base hypotheses $\mathcal{H}$, such that if one uses only T of the base hypotheses in $\mathcal{H}$, then one cannot obtain a gap smaller than $\Omega(\sqrt{\lg(n/T)/T})$. Similarly to the approach taken in [Nie et al., 2013], we model a data set $D = \{(x_i, y_i)\}_{i=1}^n$ of n data points and a corresponding set of k base hypotheses $\mathcal{H} = \{h_1, \dots, h_k\}$ as an $n \times k$ matrix A. The entry $a_{i,j}$ is equal to $y_i h_j(x_i)$. We prove our lower bound for binary classification where the hypotheses take values only amongst $\{-1,+1\}$, meaning that $A \in \{-1,+1\}^{n\times k}$. Thus an entry $a_{i,j}$ is $+1$ if hypothesis h_j is correct on point x_i and it is -1 otherwise. We remark that proving the lower bound under the restriction that $h_j(x_i)$ is among $\{-1,+1\}$ instead of $[-1,+1]$ only strengthens the lower bound.

Notice that if $w \in \mathbb{R}^k$ is a vector with $\|w\|_1 = 1$, then $(Aw)_i$ gives exactly the margin on data point (x_i, y_i) when using the linear combination $\sum_j w_j h_j$ of base hypotheses. The optimal minimum margin ρ^* for a matrix A is thus equal to $\rho^* := \max_{w\in\mathbb{R}^k:\|w\|_1=1} \min_i (Aw)_i$. We now seek a matrix A for which ρ^* is at least $\Omega(\sqrt{\lg(n/T)/T})$ larger than $\min_i(Aw)_i$ for all w with $\|w\|_0 \le T$ and $\|w\|_1 = 1$. If we can find such a matrix, it implies the existence of a data set (rows) and a set of base hypotheses (columns) for which any linear combination of up to T base hypotheses has a gap of $\Omega(\sqrt{\lg(n/T)/T})$. The lower bound thus holds regardless of how an algorithm would try to determine which linear combination to construct.

When showing the existence of a matrix A with a large gap, we will fix $k = n$, i.e. the set of base hypotheses $\mathcal{H}$ has cardinality equal to the number of data points. The following theorem shows the existence of the desired matrix A:

Theorem 3.1. *There exists a universal constant $C > 0$ such that for all sufficiently large n and all T with $\ln n \leq T \leq n/C$, there exists a matrix $A \in \{-1,+1\}^{n \times n}$ such that: 1) Let $v \in \mathbb{R}^n$ be the vector with all coordinates equal to $1/n$. Then all coordinates of Av are greater than or equal to $-O(1/\sqrt{n})$. 2) For every vector $w \in \mathbb{R}^n$ with $\|w\|_0 \leq T$ and $\|w\|_1 = 1$, it holds that:* $\min_i (Aw)_i \leq -\Omega\left(\sqrt{\lg(n/T)/T}\right)$.

Quite surprisingly, Theorem 3.1 shows that for any T with $\ln n \leq T \leq n/C$, there is a matrix $A \in \{-1,+1\}^{n \times n}$ for which the *uniform* combination of base hypotheses $\sum_{j=1}^n h_j/n$ has a minimum margin that is much higher than anything that can be obtained using only T base hypotheses. More concretely, let A be a matrix satisfying the properties of Theorem 3.1 and let $v \in \mathbb{R}^n$ be the vector with all coordinates $1/n$. Then $\rho^* := \max_{w \in \mathbb{R}^k : \|w\|_1 = 1} \min_i (Aw)_i \geq \min_i (Av)_i = -O(1/\sqrt{n})$. By the second property in Theorem 3.1, it follows that any linear combination of at most T base hypotheses must have a gap of $-O(1/\sqrt{n}) - \left(-\Omega\left(\sqrt{\lg(n/T)/T}\right)\right) = \Omega\left(\sqrt{\lg(n/T)/T}\right)$. This is precisely the claimed lower bound and also shows that our vector sparsification algorithm from Theorem 2.2 is optimal for any $T \leq n/C$. To prove Theorem 3.1, we first show the existence of a matrix $B \in \{-1,+1\}^{n \times n}$ having the second property. We then apply Spencer's Theorem (Theorem 2.3) to "transform" B into a matrix A having both properties. We find it quite surprising that Spencer's discrepancy minimization result finds applications in both our upper and lower bound.

That a matrix satisfying the second property in Theorem 3.1 exists is expressed in the following lemma:

Lemma 3.1. *There exists a universal constant $C > 0$ such that for all sufficiently large n and all T with $\ln n \leq T \leq n/C$, there exists a matrix $A \in \{-1,+1\}^{n \times n}$ such that for every vector $w \in \mathbb{R}^n$ with $\|w\|_0 \leq T$ and $\|w\|_1 = 1$ it holds that:* $\min_i (Aw)_i \leq -\Omega\left(\sqrt{\ln(n/T)/T}\right)$.

We prove Lemma 3.1 in later subsection and move on to show how we use it in combination with Spencer's Theorem to prove Theorem 3.1:

Proof. Let B be a matrix satisfying the statement in Lemma 3.1. Using Spencer's Theorem (Theorem 2.3), we get that there exists a vector $x \in \{-1,+1\}^n$ such that $\|Bx\|_\infty = O(\sqrt{n \ln(en/n)}) = O(\sqrt{n})$. Now form the matrix A which is equal to B, except that the i'th column is scaled by x_i. Then $A\mathbf{1} = Bx$ where $\mathbf{1}$ is the all-ones vectors. Normalizing the all-ones vector by a factor $1/n$ yields the vector v with all coordinates equal to $1/n$. Moreover, it holds that $\|Av\|_\infty = \|Bx\|_\infty/n = O(1/\sqrt{n})$, which in turn implies that $\min_i (Av)_i \geq -O(1/\sqrt{n})$.

Now consider any vector $w \in \mathbb{R}^n$ with $\|w\|_0 \leq T$ and $\|w\|_1 = 1$. Let $\tilde{w}$ be the vector obtained from w by multiplying w_i by x_i. Then $Aw = B\tilde{w}$. Furthermore $\|\tilde{w}\|_1 = \|w\|_1 = 1$ and $\|\tilde{w}\|_0 = \|w\|_0 \leq T$. It follows from Lemma 3.1 and our choice of B that $\min_i (Aw)_i = \min_i (B\tilde{w})_i \leq -\Omega\left(\sqrt{\ln(n/T)/T}\right)$. ☐

The proof of Lemma 3.1 is deferred to the following subsections, here we sketch the ideas. At a high level, our proof goes as follows: First argue that for a fixed vector $w \in \mathbb{R}^n$ with $\|w\|_0 \leq T$ and $\|w\|_1 = 1$ and a random matrix $A \in \{-1,+1\}^{n \times n}$ with each coordinate chosen uniformly and independently, it holds with very high probability that Aw has many coordinates that are less than $-a_1(\sqrt{\ln(n/T)/T})$ for a constant a_1. Now intuitively we would like to union bound over all possible vectors w and argue that with non-zero probability, all of them satisfies this simulatenously. This is not directly possible as there are infinitely many w. Instead, we create a *net* W consisting of a collection of carefully chosen vectors. The net will have the property that any w with $\|w\|_1 = 1$ and $\|w\|_0 \leq T$ will be close to a vector $\tilde{w} \in W$. Since the net is not too large, we can union bound over all vectors in W and find a matrix A with the above property for all vectors in W simultaneously.

For an arbitrary vector w with $\|w\|_1 = 1$ and $\|w\|_0 \leq T$, we can then write $Aw = A\tilde{w} + A(w - \tilde{w})$ where $\tilde{w} \in W$ is close to w. Since $\tilde{w} \in W$ we get that $A\tilde{w}$ has many coordinates that are less than $-a_1(\sqrt{\ln(n/T)/T})$. The problem is that $A(w - \tilde{w})$ might cancel out these negative coordinates. However, $w - \tilde{w}$ is a very short vector so this seems unlikely. To prove it formally, we further show that for every vector $v \in W$, there are also few coordinates in Av that are greater than $a_2(\sqrt{\ln(n/T)/T})$ in absolute value for some constant $a_2 > a_1$. We can then round $(w - \tilde{w})/\|w - \tilde{w}\|_1$ to a vector in the net, apply this reasoning and recurse on the difference between $(w - \tilde{w})/\|w - \tilde{w}\|_1$ and the net vector. We give the full details in the following subsections.

3.1 Preliminaries

In the following, we will introduce a few tail bounds that will be necessary in our proof of Lemma 3.1.

Definition 1. For a vector $w \in \mathbb{R}^n$, define $F(w,t)$ as follows: Let i_j be the index such that w_{i_j} is the j'th largest coordinate of w in terms of absolute value. Then:

$$F(w,t) := \left(\sum_{j=1}^{\lfloor t^2 \rfloor} |w_{i_j}| + t \left(\sum_{j=\lfloor t^2 \rfloor+1}^{n} w_{i_j}^2 \right)^{1/2} \right).$$

We need the following tail upper and lower bounds for the distribution of $\langle a, x\rangle$ where $a \in \mathbb{R}^n$ and $x \in \{-1,+1\}^n$ has uniform random and independent Rademacher coordinates:

Theorem 3.2 ([Montgomery-Smith, 1990])**.** *There exists universal constants $c_1, c_2, c_3 > 0$ such that the following holds: For any vector $w \in \mathbb{R}^n$, if $x \in \{-1,+1\}^n$ has uniform random and independent Rademacher coordinates, then:*

$$\forall t > 0 : \Pr\left[\langle w, x\rangle > c_1 F(w,t)\right] \le e^{-t^2/2}$$

and

$$\forall t > 0 : \Pr\left[\langle w, x\rangle > c_2 F(w,t)\right] \ge c_3^{-1} e^{-c_3 t^2}.$$

Lemma 3.2. *For any vector $w \in \mathbb{R}^n$, integer $t > 0$ and any $a \ge 1$, it holds that:*

$$F(w, at) \le a^2 F(w,t)$$

Proof. By definition, we see that:

$$\begin{aligned}
F(w,at) &= \left(\sum_{j=1}^{\lfloor a^2t^2 \rfloor} |w_{i_j}| + at \left(\sum_{j=\lfloor a^2t^2 \rfloor+1}^{n} w_{i_j}^2 \right)^{1/2} \right) \\
&\le \left(a^2 \sum_{j=1}^{\lfloor t^2 \rfloor} |w_{i_j}| + a^2 t \left(\sum_{j=\lfloor a^2t^2 \rfloor+1}^{n} w_{i_j}^2 \right)^{1/2} \right) \\
&\le \left(a^2 \sum_{j=1}^{\lfloor t^2 \rfloor} |w_{i_j}| + a^2 t \left(\sum_{j=\lfloor t^2 \rfloor+1}^{n} w_{i_j}^2 \right)^{1/2} \right) \\
&= a^2 F(w,t).
\end{aligned}$$

□

Corollary 3.1. *There exists a universal constant $c_4 > 0$ such that the following holds: Let A be a $k \times n$ matrix with entries independent and uniform random amongst $\{-1,+1\}$. For any vector $w \in \mathbb{R}^n$ and any integer $t \ge 4$, we have:*

$$\Pr\left[\|Aw\|_1 > c_4 k F(w,t)\right] \le e^{-t^2 k/4}.$$

Proof. Let $t \ge 4$ be integer. For each way of choosing k non-negative integers $z_1, \dots, z_k$ such that $\sum_i z_i = k$, define an event $E_{z_1,\dots,z_k}$ that happens when:

$$\forall h : |(Aw)_h| \ge c_1 z_h F(w,t).$$

We wish to bound $\Pr[E_{z_1,\dots,z_k}]$. Using Lemma 3.2 we get:

$$|(Aw)_h| \geq c_1 z_h F(w,t) \quad \Rightarrow$$
$$|(Aw)_h| \geq c_1 F(w, \sqrt{z_h} t),$$

whenever $z_h \geq 1$. Using that the distribution of $(Aw)_h$ is symmetric around 0, we get from Theorem 3.2 that

$$\Pr[E_{z_1,\dots,z_k}] \leq \prod_{h=1}^{k} 2e^{-(\sqrt{z_h} t)^2/2} = 2^k e^{-kt^2/2}.$$

Let $c_4 = 2c_1$ where c_1 is the constant from Theorem 3.2. If

$$\|Aw\|_1 > c_4 k F(w,t)$$

then at least one of the events $E_{z_1,\dots,z_k}$ must happen. Since there are $\binom{k+k-1}{k} \leq 2^{2k}$ such events, we conclude from a union bound that for integer $t \geq 4$, we have

$$\Pr\left[\|Aw\|_1 > c_4 k F(w,t)\right] \leq 2^{3k} e^{-kt^2/2} < e^{-k(t^2/2-3)} \leq e^{-k(t^2/2-t^2/4)} = e^{-kt^2/4}.$$

□

We will also need the Chernoff bound:

Theorem 3.3 (Chernoff Bound)**.** *Let $X_1, \dots, X_n$ be independent $\{0,1\}$-random variables and let $X = \sum_i X_i$. Then for any $0 < \delta < 1$:*

$$\Pr[X \leq (1-\delta)\mathbb{E}[X]] \leq e^{-\delta^2 \mathbb{E}[X]/2}$$

3.2 The Net Argument

We are ready to prove Lemma 3.1. As highlighted above, our proof will use a net argument. What we wanted to construct, was a collection of vectors W, such that any vector w with $\|w\|_0 \leq T$ and $\|w\|_1 = 1$ was close to some vector in W, and moreover, for a random matrix $A \in \{-1,+1\}^{n \times n}$, there is a non-zero probability that every vector $w \in W$ satisfies that there are many coordinates in Aw that are less than $-a_1(\sqrt{\lg(n/T)/T})$, but few that are greater than $a_2(\sqrt{\lg(n/T)/T})$ in absolute value for some constants $a_1 < a_2$. We formulate this property in the following definition:

Definition 2. Let $A \in \mathbb{R}^{n \times n}$ be an $n \times n$ real matrix and let $w \in \mathbb{R}^n$. We say that A is (t,b,T)-*typical* for w if the following two holds:

- $|\{i : (Aw)_i < -c_2 F(w,t)\}| \geq \ln \binom{n}{T}$, where c_2 is the constant from Theorem 3.2.
- For all sets of $k = \ln \binom{n}{T}$ rows of A, it holds that the corresponding $k \times n$ submatrix $\bar{A}$ satisfies $\|\bar{A}w\|_1 < bkF(w,t)$.

Observe that the second property in the definition says that there cannot be too many very large coordinates in Aw. The following lemma shows that for a fixed vector $w \in \mathbb{R}^n$ and a random matrix A, there is a very high probability that A is typical for w:

Lemma 3.3. *There exists universal constants $b_1, b_2 > 0$ such that the following holds for all n sufficiently large: Let A be a random $n \times n$ matrix with each entry chosen independently and uniformly at random amongst $\{-1,+1\}$. Define*

$$\tau(T) := \left\lfloor \sqrt{c_3^{-1} \ln\left(\frac{n}{12 c_3 \ln \binom{n}{T}}\right)} \right\rfloor$$

where c_3 *is the constant from Theorem 3.2. Then the following holds for all* $T \leq n/b_1$*: If* $w \in \mathbb{R}^n$ *with* $\|w\|_0 \leq T$*, then:*

$$\Pr[A \text{ is } (\tau(m), b_2, T)\text{-typical for } w] \geq 1 - \binom{n}{T}^{-3}.$$

Furthermore, it holds that $\tau(T) = \Omega(\sqrt{\ln(n/T)})$.

Proof. Let a_i be the i'th row of A. Using that the distribution of $\langle a, w\rangle$ is symmetric around 0, Theorem 3.2 gives us:

$$\forall t > 0 : \Pr[\langle w, x\rangle < -c_2 F(w,t)] \geq c_3^{-1} e^{-c_3 t^2}$$

Defining

$$\tau(T) := \left\lfloor \sqrt{c_3^{-1} \ln\left(\frac{n}{12 c_3 \ln\binom{n}{T}}\right)} \right\rfloor$$

gives

$$\Pr[\langle w, x\rangle < -c_2 F(w, \tau(T))] \geq c_3^{-1} e^{-c_3 \tau(T)^2} \geq \frac{12 \ln\binom{n}{T}}{n}.$$

Now define X_i to take the value 1 if $\langle w, x\rangle < -c_2 F(w, \tau(T))$ and 0 otherwise. Let $X = \sum_i X_i$. Then $\mathbb{E}[X] \geq 12 \ln\binom{n}{T}$. It follows from the Chernoff bound (Theorem 3.3) that

$$\Pr\left[X \leq \ln\binom{n}{T}\right] \leq e^{-(11/12)^2 6 \ln\binom{n}{T}} = \binom{n}{T}^{-11^2/24} < \binom{n}{T}^{-5} < \frac{\binom{n}{T}^{-3}}{2}.$$

Thus the first requirement for being $(\tau(T), b_2, T)$-typical is satisfied with probability at least

$$1 - \frac{\binom{n}{T}^{-3}}{2}.$$

Our next step is to show the same for the second property. To do this, we will be using Corollary 3.1, which requires $t \geq 4$. If we restrict $T \leq n/b_1$ for a big enough constant $b_1 > 0$, this is indeed the case for our choice of $\tau(T)$ (since we can make $\ln\binom{n}{T} \leq n/b'$ for any desirable constant $b' > 0$ by setting b_1 to a large enough constant). Thus from here on, we fix b_1 to a constant large enough that $\tau(T) \geq 4$ whenever $T \leq n/b_1$.

Now fix a set of $k = \ln\binom{n}{T}$ rows of A and let $\bar{A}$ denote the corresponding $k \times n$ submatrix. Corollary 3.1 gives us that $\Pr[\|\bar{A}w\|_1 > c_4 k F(w, \alpha\tau(T))] \leq e^{-\alpha^2 \tau(T)^2 k/4} = \binom{n}{T}^{-\alpha^2 \tau(T)^2/4}$ for any constant $\alpha \geq 1$. If we set α to a large enough constant, we get that $\alpha^2 \tau(T)^2/4 \geq 5 \ln(en/k)$. Thus $\Pr[\|\bar{A}w\|_1 > c_4 k F(w, \alpha\tau(T))] \leq e^{-5k \ln(en/k)}$. There are $\binom{n}{k} \leq e^{k \ln(en/k)}$ sets of k rows in A, thus by a union bound, we get that with probability at least $1 - e^{-4k \ln(en/k)} \geq 1 - \binom{n}{T}^{-3}/2$, it simultanously holds that $\|\bar{A}w\|_1 \leq c_4 k F(w, \alpha\tau(T))$ for all $k \times n$ submatrices $\bar{A}$ of A. From Lemma 3.2, this implies $\|\bar{A}w\|_1 \leq c_4 \alpha^2 k F(w, \tau(T))$. Another union bound shows that A is $(\tau(T), b_2, T)$-typical for w with probability at least $1 - \binom{n}{T}^{-3}$ if we set $b_2 = c_4 \alpha^2$.

What remains is to show that $\tau(T) = \Omega(\sqrt{\ln(n/T)})$. We see that:

$$\left\lfloor \sqrt{c_3^{-1} \ln\left(\frac{n}{12 c_3 \ln\binom{n}{T}}\right)} \right\rfloor \quad \geq \quad \left\lfloor \sqrt{c_3^{-1} \ln\left(\frac{n}{12 c_3 T \ln(en/T)}\right)} \right\rfloor$$

If b_1 is a large enough constant (i.e. T small enough compared to n), then $\ln(en/T) \leq 2\ln(n/T)$. Thus for large enough b_1:

$$\tau(T) \quad \geq \quad \left\lfloor \sqrt{c_3^{-1}(\ln(n/T) - \ln\ln(n/T) - \ln(24 c_3))} \right\rfloor$$

the terms $\ln(\ln(n/T))$ and $\ln(24c_3)$ are at most small constant factors of $\ln(n/T)$ and the claim follows for b_1 big enough. ∎

We are ready to define the net W which we will union bound over and show that a random matrix is typical for all vectors in W simultanously with non-zero probability. Let b_1, b_2 be the constants from Lemma 3.3 and let $T \leq n/b_1 b_3^2$ with $b_3 > b_2$ a large constant. Construct a collection W of vectors in $\mathbb{R}^n$ as follows: For every choice of m distinct coordinates $i_1, \ldots, i_T \in [n]$, and every choice of T integers $z_1, \ldots, z_T$ with $\sum_j |z_j| \leq (b_3+1)T$, add the vector w with coordinates $w_{i_j} = z_j/(b_3 T)$ for $j = 1, \ldots, T$ and $w_h = 0$ for $h \notin \{i_1, \ldots, i_T\}$ to W. We have

$$|W| \leq \binom{n}{T} 2^T \sum_{i=0}^{(b_3+1)T} \binom{T+i-1}{T}.$$

To see this, observe that $\binom{n}{T}$ counts the number of ways to choose $i_1, \ldots, i_T$, 2^T counts the number of ways of choosing the signs of the z_j's and $\binom{T+i-1}{T}$ counts the number of ways to choose T non-negative integers summing to i.

If b_3 is large enough, then $(b_3+1)T \leq (b_3+1)n/(b_1 b_3^2)$ is much smaller than n and hence $|W| < \binom{n}{T}^3$. Since $T < n/b_1$, we can use Lemma 3.3 and a union bound over all $w \in W$ to conclude that there exists an $n \times n$ matrix A with coordinates amongst $\{-1, +1\}$, such that A is $(\tau(T), b_2, T)$-typical for all $w \in W$ with $\tau(T) = \Omega(\sqrt{\ln(n/T)})$. We claim that this implies the following:

Lemma 3.4. *Let $A \in \{-1,+1\}^{n \times n}$ be $(\tau(T), b_2, T)$-typical for all $w \in W$. If $\ln n \leq T$, then for any vector x with $\|x\|_0 \leq T$ and $\|x\|_1 = 1$, there exists an index i such that*

$$(Ax)_i \leq -\Omega\left(\sqrt{\frac{\ln(n/T)}{T}}\right).$$

Proof. Let x be as in the lemma, i.e. $\|x\|_0 \leq T$ and $\|x\|_1 = 1$. To prove the lemma, we will "round" the arbitrary vector x to a close-by vector in W, allowing us to use that A is typical for all vectors in W.

For a vector x with $\|x\|_0 \leq T$ and $\|x\|_1 = 1$, let $f(x) \in W$ be the vector in W obtained by rounding each coordinate of x up (in absolute value) to the nearest multiple of $1/(b_3 T)$. The vector $f(x)$ is in W since $\sum_i |f(x)_i| \cdot (b_3 T) \leq (b_3 T)(1 + T/b_3 T) \leq b_3 T + T = (b_3+1)T$ and that each coordinate is an integer multiple of $1/(b_3 T)$.

Notice that the vector $\tilde{x} = x - f(x)$ has at most T non-zero coordinates, all between 0 and $1/(b_3 T)$ in absolute value. Now observe that:

$$\begin{aligned} Ax &= A(f(x) + \tilde{x}) \\ &= Af(x) + A\tilde{x}. \end{aligned}$$

Since $f(x) \in W$, we have that A is $(\tau(T), b_2, T)$-typical for $f(x)$. Hence there exists at least $\ln \binom{n}{T}$ coordinates of $Af(x)$ such that $(Af(x))_i < -c_2 F(f(x), \tau(T))$. If we let i_j denote the index of the j'th largest coordinate of $f(x)$ in terms of absolute value, then we have:

$$F(f(x), \tau(T)) = \left(\sum_{j=1}^{\lfloor \tau(T)^2 \rfloor} |f(x)_{i_j}| + \tau(T) \left(\sum_{j=\lfloor \tau(T)^2 \rfloor + 1}^{n} f(x)_{i_j}^2 \right)^{1/2} \right).$$

By construction, we have that $\|f(x)\|_1 \geq \|x\|_1 = 1$ (f rounds coordinates up in absolute value). This means that either $\sum_{j=1}^{\lfloor \tau(T)^2 \rfloor} |f(x)_{i_j}| \geq 1/2$ or $\sum_{j=\lfloor \tau(T)^2 \rfloor+1}^{n} |f(x)_{i_j}| \geq 1/2$. In the first case, we have $F(f(x), \tau(T)) \geq 1/2$. In the latter case, we use Cauchy-Schwartz to conclude that

$$\left(\sum_{j=\lfloor \tau(T)^2 \rfloor+1}^{n} f(x)_{i_j}^2 \right)^{1/2} \geq (1/2)/\|f(x)\|_0^{1/2} \geq 1/(2\sqrt{T}).$$

To see how this follows from Cauchy-Schwartz, define $\overline{f(x)}$ as the vector having the same coordinates as $f(x)$ for indices i_j with $j \geq \lfloor \tau(T)^2 \rfloor + 1$ and 0 elsewhere. Define y as the vector with a 1 in all coordinates i where $\overline{f(x)}_i \neq 0$ and 0 elsewhere. Then by Cauchy-Schwartz

$$1/2 \leq \sum_i |y_i \overline{f(x)}_i| \leq \sqrt{\|\overline{f(x)}\|_2 \|y\|_2} \leq \sqrt{\|\overline{f(x)}\|_2 \|f(x)\|_0}.$$

Hence we must have

$$F(f(x), \tau(T)) \geq \min\left\{1/2, \frac{\tau(T)}{\sqrt{T}}\right\}.$$

Therefore, we have at least $\ln \binom{n}{T}$ coordinates i of $Af(x)$ with

$$(Af(x))_i \leq -c_2 \min\left\{1/2, \frac{\tau(T)}{\sqrt{T}}\right\},$$

where c_2 is the constant from Theorem 3.2.

We let T denote an arbitrary subset of $\ln \binom{n}{T}$ such indices. What remains is to show that $A\tilde{x}$ cannot cancel out these very negative coordinates. To prove this, we will write $\tilde{x}$ as a sum $\|\tilde{x}\|_1 \cdot \sum_{j=0}^{\infty} \alpha_j \tilde{x}^{(j)}$ with each $\tilde{x}^{(j)} \in W$. The idea is to repeatedly apply the function f to find a vector in W that is close to $\tilde{x}$. We then append that to the sum and recurse on the "rounding error" $\tilde{x} - f(\tilde{x})$. To do this more formally, define

$$\begin{aligned} \hat{x}^{(0)} &:= \tilde{x}/\|\tilde{x}\|_1 \\ \tilde{x}^{(0)} &:= f(\hat{x}^{(0)}) \\ \alpha_0 &:= 1 \end{aligned}$$

and for $j > 0$, define

$$\begin{aligned} \hat{x}^{(j)} &:= \hat{x}^{(j-1)} - \alpha_{j-1}\tilde{x}^{(j-1)} \\ \tilde{x}^{(j)} &:= f(\hat{x}^{(j)}/\|\hat{x}^{(j)}\|_1) \\ \alpha_j &:= \|\hat{x}^{(j)}\|_1. \end{aligned}$$

We would like to show that $\tilde{x} = \|\tilde{x}\|_1 \cdot \sum_{j=0}^{\infty} \alpha_j \tilde{x}^{(j)}$. The first step in proving this, is to bound α_j and $\hat{x}^{(j)}$ as j tends to infinity. We see that: $\alpha_j = \|\hat{x}^{(j)}\|_1$. For $j \neq 0$, we therefore get:

$$\begin{aligned} \alpha_j &= \|\hat{x}^{(j-1)} - \alpha_{j-1}\tilde{x}^{(j-1)}\|_1 \\ &= \|\hat{x}^{(j-1)} - \alpha_{j-1} f(\hat{x}^{(j-1)}/\alpha_{j-1})\|_1 \\ &= \alpha_{j-1}\|\hat{x}^{(j-1)}/\alpha_{j-1} - f(\hat{x}^{(j-1)}/\alpha_{j-1})\|_1 \\ &\leq \alpha_{j-1} T(1/b_3 T) \\ &= \alpha_{j-1}/b_3. \end{aligned}$$

This implies that $\lim_{j\to\infty} \alpha_j = 0$. Since $\alpha_j = \|\hat{x}^{(j)}\|_1$, this also implies that $\lim_{j\to\infty} \hat{x}^{(j)} = 0$. We can now conclude that:

$$\begin{aligned}
\|\tilde{x}\|_1 \cdot \sum_{j=0}^{\infty} \alpha_j \tilde{x}^{(j)} &= \\
\|\tilde{x}\|_1 \cdot \left(f(\hat{x}^{(0)}) + \sum_{j=1}^{\infty} \alpha_j \left(\frac{\hat{x}^{(j)} - \hat{x}^{(j+1)}}{\alpha_j} \right) \right) &= \\
\|\tilde{x}\|_1 \cdot \left(f(\hat{x}^{(0)}) + \sum_{j=1}^{\infty} \hat{x}^{(j)} - \hat{x}^{(j+1)} \right) &= \\
\|\tilde{x}\|_1 \cdot \left(f(\hat{x}^{(0)}) + \left(\hat{x}^{(1)} - \lim_{j\to\infty} \hat{x}^{(j)} \right) \right) &= \\
\|\tilde{x}\|_1 \cdot \left(f(\hat{x}^{(0)}) + \left(\hat{x}^{(0)} - \tilde{x}^{(0)} \right) \right) &= \\
\|\tilde{x}\|_1 \cdot \left(f(\hat{x}^{(0)}) + \left(\hat{x}^{(0)} - f(\hat{x}^{(0)}) \right) \right) &= \\
\tilde{x}.
\end{aligned}$$

Thus we have that $A\tilde{x} = A\left(\|\tilde{x}\|_1 \sum_{j=0}^{\infty} \alpha_j \tilde{x}^{(j)} \right)$. Now let $\bar{A}$ be the $\ln \binom{n}{T} \times n$ submatrix of A corresponding to the rows in the set T. We wish to bound $\|\bar{A}\tilde{x}\|_1$. Using the triangle inequality and that A is $(\tau(T), b_2, T)$-typical for all vectors in W, we get:

$$\|\bar{A}\tilde{x}\|_1 \leq \|\tilde{x}\|_1 \sum_{j=0}^{\infty} \alpha_j \|\bar{A}\tilde{x}^{(j)}\|_1 \leq \|\tilde{x}\|_1 \ln \binom{n}{T} b_2 \sum_{j=0}^{\infty} \alpha_j F(\tilde{x}^{(j)}, \tau(T)).$$

We want to argue that $F(\tilde{x}^{(j)}, \tau(T))$ is small. For this, we will bound the magnitude of coordinates in $\tilde{x}^{(j)}$. We start with $j \neq 0$, in which case we have $\tilde{x}^{(j)} = f(\hat{x}^{(j)}/\|\hat{x}^{(j)}\|_1)$. By definition of f, this means that $\|\tilde{x}^{(j)}\|_\infty \leq \|\hat{x}^{(j)}/\|\hat{x}^{(j)}\|_1\|_\infty + 1/b_3T$. Using that

$$\begin{aligned}
\hat{x}^{(j)} &= \hat{x}^{(j-1)} - \alpha_{j-1}\tilde{x}^{(j-1)} \\
&= \hat{x}^{(j-1)} - \alpha_{j-1} f(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1) \\
&= \hat{x}^{(j-1)} - \|\hat{x}^{(j-1)}\|_1 f(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1) \\
&= \|\hat{x}^{(j-1)}\|_1 \left(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1 - f(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1) \right),
\end{aligned}$$

we get by definition of f that $\left(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1 - f(\hat{x}^{(j-1)}/\|\hat{x}^{(j-1)}\|_1)\right)$ has at most T non-zero coordinates, all between 0 and $1/b_3T$ in absolute value. We have thus shown that

$$\|\tilde{x}^{(j)}\|_\infty \leq \|\hat{x}^{(j)}/\|\hat{x}^{(j)}\|_1\|_\infty + 1/b_3T \leq \left(1 + \frac{\|\hat{x}^{(j-1)}\|_1}{\|\hat{x}^{(j)}\|_1}\right) \cdot \frac{1}{b_3T} = \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T}.$$

We therefore have for $j \neq 0$ that:

$$\begin{aligned}
F(\tilde{x}^{(j)}, \tau(m)) &\leq \\
\left(\sum_{j=1}^{\lfloor \tau(T)^2 \rfloor} \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T} + \tau(T) \left(\sum_{j=\lfloor \tau(T)^2 \rfloor + 1}^{T} \left(\left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T} \right)^2 \right)^{1/2} \right) &\leq \\
\tau(T)^2 \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T} + \tau(T)\sqrt{T} \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T} &= \\
\tau(T)^2 \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3T} + \tau(T) \left(1 + \frac{\alpha_{j-1}}{\alpha_j}\right) \cdot \frac{1}{b_3\sqrt{T}}.
\end{aligned}$$

For $j = 0$, we see that

$$\|\tilde{x}^{(0)}\|_\infty = \|f(\hat{x}^{(0)}\|_\infty \le \|\tilde{x}/\|\tilde{x}\|_1\|_\infty + 1/b_3T = \|\tilde{x}\|_\infty/\|\tilde{x}\|_1 + 1/b_3T.$$

But $\tilde{x}$ was the difference between x and $f(x)$ and therefore $\|\tilde{x}\|_\infty \le 1/b_3T$ and we get:

$$\|\tilde{x}^{(0)}\|_\infty \le \left(1 + \frac{1}{\|\tilde{x}\|_1}\right)/b_3T.$$

We then have:

$$F(\tilde{x}^{(0)}, \tau(T)) \quad \le \quad \tau(T)^2\left(1 + \frac{1}{\|\tilde{x}\|_1}\right) \cdot \frac{1}{b_3T} + \tau(T)\left(1 + \frac{1}{\|\tilde{x}\|_1}\right) \cdot \frac{1}{b_3\sqrt{T}},$$

which allows us to conclude:

$$\begin{aligned}
\|\bar{A}\tilde{x}\|_1 \quad &\le \quad \|\tilde{x}\|_1 \ln\binom{n}{T} b_2 \sum_{j=0}^{\infty} \alpha_j F(\tilde{x}^{(j)}, \tau(T)) \\
&= \quad \|\tilde{x}\|_1 \ln\binom{n}{T} b_2 \cdot \left(\frac{\tau(T)^2}{b_3T} + \frac{\tau(T)}{b_3\sqrt{T}}\right) \cdot \left(\left(1 + \frac{1}{\|\tilde{x}\|_1}\right) + \sum_{j=1}^{\infty}(\alpha_{j-1} + \alpha_j)\right) \\
&= \quad \|\tilde{x}\|_1 \ln\binom{n}{T} b_2 \cdot \left(\frac{\tau(T)^2}{b_3T} + \frac{\tau(T)}{b_3\sqrt{T}}\right) \cdot \left(\frac{1}{\|\tilde{x}\|_1} + 2 \cdot \sum_{j=0}^{\infty}\alpha_j\right).
\end{aligned}$$

Recalling that $\alpha_j \le \alpha_{j-1}/b_3$, we can write: $\sum_{j=0}^{\infty}\alpha_j \le \sum_{j=0}^{\infty}(1/b_3)^j \le 2$ for $b_3 \ge 2$. Using that $\tilde{x}$ has at most T non-zero coordinates, all of absolute value at most $1/b_3T$, we also have $\|\tilde{x}\|_1 \le 1/b_3$. Therefore we conclude:

$$\|\bar{A}\tilde{x}\|_1 \quad \le \quad \ln\binom{n}{T} b_2 \left(\frac{\tau(T)^2}{b_3T} + \frac{\tau(T)}{b_3\sqrt{T}}\right) \cdot (1 + 4/b_3)\,.$$

Hence there must be one of the coordinates i where

$$|(\bar{A}\tilde{x})_i| \le \frac{b_2}{b_3}\left(\frac{\tau(T)^2}{T} + \frac{\tau(T)}{\sqrt{T}}\right) \cdot (1 + 4/b_3)\,.$$

But all $\ln\binom{n}{T}$ coordinates of $\bar{A}f(x)$ were less than

$$-c_2 \min\left\{1/2, \frac{\tau(T)}{\sqrt{T}}\right\}$$

thus for b_3 a large enough constant and $\tau(T) \le \sqrt{T}$, there must be a coordinate i where

$$(Ax)_i \le -\Omega\left(\min\left\{1, \sqrt{\frac{\ln(n/T)}{T}}\right\}\right).$$

Under the assumption that $\tau(T) \le \sqrt{T} \Leftrightarrow \ln(n/T) \le T \Leftarrow \ln n \le T$, this simplifies to:

$$(Ax)_i \le -\Omega\left(\sqrt{\frac{\ln(n/T)}{T}}\right).$$

□

We have argued that the random matrix A was typical for all vectors in W simultanously with non-zero probability. Combining this with Lemma 3.4 finally concludes the proof of Lemma 3.1. g

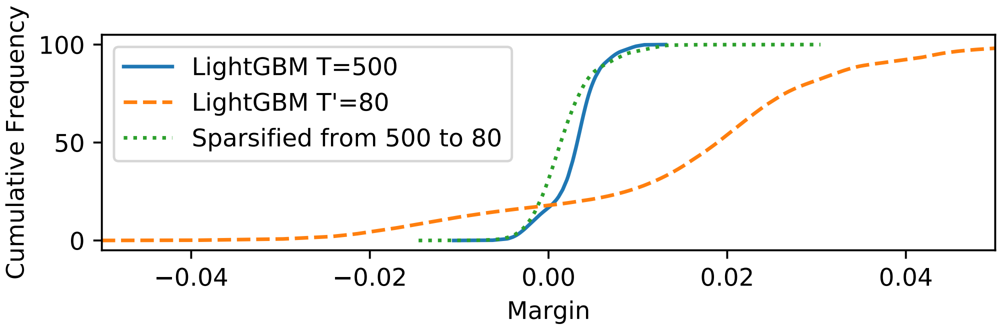

Figure 1: The plot depicts the cumulative margins of three classifiers: (1) a LightGBM classifier with 500 hypotheses (2) a classifier sparsified from 500 to 80 hypotheses and (3) a LightGBM classifier with 80 hypotheses.

4 Experiments

Gradient Boosting [Mason et al., 2000, Friedman, 2001] is probably the most popular boosting algorithm in practice. It has several highly efficient open-source implementations [Chen and Guestrin, 2016, Ke et al., 2017, Prokhorenkova et al., 2018] and obtain state-of-the-art performance in many machine learning tasks [Ke et al., 2017]. In this section we demonstrate how our sparsification algorithm can be combined with Gradient Boosting. For simplicity we consider a single dataset in this section, the Flight Delay dataset [air,], see Appendix B for similar results on other dataset.

We train a classifier with $T = 500$ hypotheses using LightGBM [Ke et al., 2017] which we sparsify using Theorem 2.2 to have $T' = 80$ hypotheses. The sparsified classifier is guaranteed to preserve all margins of the original classifier to an additive $O(\sqrt{\lg(n/T')/T'})$. The cumulative margins of the sparsified classifier and the original classifier are depicted in Figure 1. Furthermore, we also depict the cumulative margins of a LightGBM classifier trained to have $T' = 80$ hypotheses. First observe the difference between the LightGBM classifiers with $T = 500$ and $T' = 80$ hypotheses (blue and orange in Figure 1). The margins of the classifier with $T = 500$ hypotheses vary less. It has fewer points with a large margin, but also fewer points with a small margin. The margin distribution of the sparsified classifier with $T' = 80$ approximates the margin distribution of the LightGBM classifier with $T = 500$ hypotheses. Inspired by margin theory one might suspect this leads to better generalization. To investigate this, we performed additional experiments computing AUC and classification accuracy of several sparsified classifiers and LightGBM classifiers on a test set (we show the results for multiple sparsified classifiers due to the randomization in the discrepancy minimization algorithms). The experiments indeed show that the sparsified classifiers outperform the LightGBM classifiers with the same number of hypotheses. See Figure 2 for test AUC and test classification accuracy.

Further Experiments and Importance Sampling. Inspired by the experiments in [Wang et al., 2008, Ke et al., 2017, Chen and Guestrin, 2016] we also performed the above experiments on the Higgs [Whiteson, 2014] and Letter [Dheeru and Karra Taniskidou, 2017] datasets. See Appendix B for (1) further experimental details and (2) for cumulative margin, AUC and test accuracy plots on all dataset for different values of n and T.

As mentioned in Section 2, one could use importance sampling for sparsification. It has a slightly worse theoretical guarantee, but might work better in practice. Appendix B also contains test AUC and test accuracy of the classifiers that result from using importance sampling instead of our algorithm based on discrepancy minimization. Our algorithm and importance sampling are both random so the experiments were repeated several times. On average over the experiments, our algorithm obtains a better test AUC and classification accuracy than importance sampling.

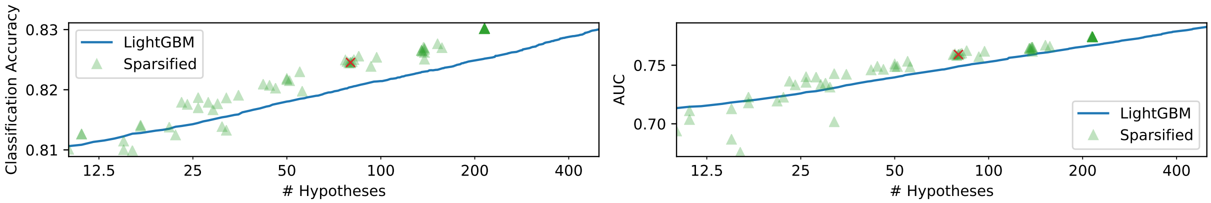

Figure 2: The plot depicts test AUC and test classification accuracy of a LightGBM classifier during training as the number of hypotheses increase (in blue). Notice the x-axis is logarithmically scaled. The final classifier with 500 hypotheses was sparsified with Theorem 2.2 multiple times to have between $T/2$ to $T/16$ hypotheses. The green triangles show test AUC and test accuracy of the resulting sparsified classifiers. The red cross represents the sparsified classifier used to plot the cumulative margins in Figure 1.

A Python/NumPy implementation of our sparsification algorithm (Theorem 2.2) can be found at:

`https://github.com/AlgoAU/DiscMin`

5 Conclusion

A long line of research into obtaining a large minimal margin using few hypotheses [Breiman, 1999, Grove and Schuurmans, 1998, Bennett et al., 2000, Rätsch and Warmuth, 2002] culminated with the AdaBoostV [Rätsch and Warmuth, 2005] algorithm. AdaBoostV was later conjectured by [Nie et al., 2013] to provide an optimal trade-off between minimal margin and number of hypotheses. In this article, we introduced SparsiBoost which refutes the conjecture of [Nie et al., 2013]. Furthermore, we show a matching lower bound, which implies that SparsiBoost is optimal.

The key idea behind SparsiBoost, is a sparsification algorithm that reduces the number of hypotheses while approximately preserving the entire margin distribution. Experimentally, we combine our sparsification algorithm with LightGBM. We find that the sparsified classifiers obtains a better margin distribution, which typically yields a better test AUC and test classification error when compared to a classifier trained directly to the same number of hypotheses.

References

[air,] Flight delay data. `https://github.com/szilard/benchm-ml#data`.

[Bennett et al., 2000] Bennett, K. P., Demiriz, A., and Shawe-Taylor, J. (2000). A column generation algorithm for boosting. In *ICML*, pages 65–72.

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

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

[Dheeru and Karra Taniskidou, 2017] Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

[Freund et al., 1999] Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

[Freund and Schapire, 1995] Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

[Gao and Zhou, 2013] Gao, W. and Zhou, Z.-H. (2013). On the doubt about margin explanation of boosting. *Artificial Intelligence*, 203:1–18.

[Grove and Schuurmans, 1998] Grove, A. J. and Schuurmans, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles. In *AAAI/IAAI*, pages 692–699.

[Ke et al., 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

[Klein and Young, 1999] Klein, P. and Young, N. (1999). On the number of iterations for dantzig-wolfe optimization and packing-covering approximation algorithms. *Lecture Notes in Computer Science*, 1610:320–327.

[Koltchinskii et al., 2001] Koltchinskii, V., Panchenko, D., and Lozano, F. (2001). Some new bounds on the generalization error of combined classifiers. In *Advances in neural information processing systems*, pages 245–251.

[Larsen, 2017] Larsen, K. G. (2017). Constructive discrepancy minimization with hereditary L2 guarantees. *CoRR*, abs/1711.02860.

[Lovett and Meka, 2015] Lovett, S. and Meka, R. (2015). Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582.

[Mason et al., 2000] Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. R. (2000). Boosting algorithms as gradient descent. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12*, pages 512–518. MIT Press.

[McKinney et al., 2010] McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

[Montgomery-Smith, 1990] Montgomery-Smith, S. (1990). The distribution of Rademacher sums. *Proc. Amer. Math. Soc.*, 109:517–522.

[Nie et al., 2013] Nie, J., Warmuth, M., Vishwanathan, S., and Zhang, X. (2013). Open problem: Lower bounds for boosting with hadamard matrices. *Journal of Machine Learning Research*, 30:1076–1079.

[Oliphant, 2006] Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Prokhorenkova et al., 2018] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6637–6647.

[Rätsch and Warmuth, 2002] Rätsch, G. and Warmuth, M. K. (2002). Maximizing the margin with boosting. In *COLT*, volume 2375, pages 334–350. Springer.

[Rätsch and Warmuth, 2005] Rätsch, G. and Warmuth, M. K. (2005). Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, 6(Dec):2131–2152.

[Reyzin and Schapire, 2006] Reyzin, L. and Schapire, R. E. (2006). How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd international conference on Machine learning*, pages 753–760. ACM.

[Schapire et al., 1998] Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., et al. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686.

[Spencer, 1985] Spencer, J. (1985). Six standard deviations suffice. *Transactions of the American mathematical society*, 289(2):679–706.

[Wang et al., 2008] Wang, L., Sugiyama, M., Yang, C., Zhou, Z.-H., and Feng, J. (2008). On the margin explanation of boosting algorithms. In *COLT*, pages 479–490. Citeseer.

[Whiteson, 2014] Whiteson, D. (2014). Higgs data set. `https://archive.ics.uci.edu/ml/datasets/HIGGS`.

A Rewriting bounds

In the following, lg is the base 2 logarithm. Here we show that if $T \leq c_1 \lg(nv^2)/v^2$ for a constant $c_1 > 0$ and $T \leq n/2$, then $v \leq c_2\sqrt{\lg(n/T)/T}$ for a constant $c_2 > 0$. We split the proof in two parts:

1. If $v^2 > 1/n$: First note that $T \leq c_1 \lg(nv^2)/v^2 \Rightarrow v \leq \sqrt{c_1 \lg(nv^2)/T}$. We now claim that $\lg(nv^2) \leq c_3 \lg(n/T)$ for a constant $c_3 > 0$. Using our bound $v \leq \sqrt{c_1 \lg(nv^2)/T}$ we get $\lg(nv^2) \leq \lg(nc_1 \lg(nv^2)/T) = \lg(n/T) + \lg c_1 + \lg\lg(nv^2)$. We have $\lg\lg(nv^2) \leq \lg(nv^2) * 0.9$ (since $\lg\lg x < (\lg x) * 0.9$ for all $x > 1$). This implies $\lg(nv^2)/10 \leq \lg(n/T) + \lg c_1$. Since $T \leq n/2$ we have $\lg(n/T) \geq 1$ and it follows that $\lg(nv^2) \leq (10 + \max\{0, 10\lg c_1\})\lg(n/T)$. Defining $c_3 = (10 + \max\{0, 10\lg c_1\})$ we thus conclude that $v \leq \sqrt{c_1 \lg(nv^2)/T} \Rightarrow v \leq \sqrt{c_1 c_3 \lg(n/T)/T}$ as claimed.

2. If $v^2 \leq 1/n$: Here we immediately have $v \leq \sqrt{\lg(n/T)/T}$ since $T \leq n/2$.

Next we show that if $v \geq c_1\sqrt{\lg(n/T)/T}$ for a constant $c_1 > 0$ and v satisfies $v > \sqrt{1/n}$, then $T \geq c_2 \lg(nv^2)/v^2$ for a constant $c_2 > 0$. We split the proof in two parts:

1. If $T \leq \lg(nv^2)/v^2$: Observe that $v \geq c_1\sqrt{\lg(n/T)/T} \Rightarrow T \geq c_1^2 \lg(n/T)/v^2$. Using the assumption $T \leq \lg(nv^2)/v^2$ we get $T \geq c_1^2 \lg(nv^2/\lg(nv^2))/v^2 = c_1^2(\lg(nv^2) - \lg\lg(nv^2))/v^2$. But $\lg\lg(nv^2) \leq \lg(nv^2) * 0.9$ since $nv^2 > 1$. Therefore we conclude $T \geq (c_1^2/10)\lg(nv^2)/v^2$.

2. If $T > \lg(nv^2)/v^2$: This case is trivial as we already satisfy the claim.

B Additional Experiments and Experimental Details

We train a LightGBM classifier and sparsify it in two ways: Theorem 2.2 and importance sampling. Both sparsification algorithms are random so we repeated both sparsifications 10 times. The experiment was performed on the following dataset:

- Inspired by the XGBoost article [Chen and Guestrin, 2016] we used the Higgs dataset [Whiteson, 2014]. We shuffled the 10^7 training examples and selected the first 10^6 examples for training and the following 10^6 points for examples.

- Inspired by the LightGBM article [Ke et al., 2017] we used the Flight Delay dataset [air,]. The dataset contain delay times and was turned into binary classification by predicting if the flight was delayed or not. All features of the 10^7 training examples were one-hot encoded with the Pandas [McKinney et al., 2010] function *get_ dummies()* which yielded 660 features. We shuffled the 10^7 training examples and selected the first 10^6 examples for training and the following 10^6 examples for testing.
- Inspired by the equivalence margin article [Wang et al., 2008] we used the Letter dataset [Dheeru and Karra Taniskidou, 2017]. It was turned into a binary classification problem as done in [Wang et al., 2008]. We shuffled the 20000 examples and used the first 10000 for training and the last 10000 for testing.

The initial classifiers were trained with LightGBM using default parameters (except for the Letter dataset where we used LightGBM decision stumps inspired by [Wang et al., 2008]). The rest of this article contain figures that show different variants our experiments. Each figure concerns a single dataset for a choice of *number of hypotheses* T and *number of points* n. For example Figure 3 contains results for our experiment on the Flight Delay dataset with $T = 100$ hypotheses and $n = 250000$ training and test points. It contains a margin plot similar to Figure 1 and test AUC/accuracy plot similar to Figure 2. See Table 1 for an overview of all experiments.

Dataset	**n**	**T**
Letter	10000	100
Flight Delay	250000	100
Flight Delay	500000	250
Flight Delay	1000000	500
Higgs	250000	100
Higgs	500000	250
Higgs	1000000	500

Table 1: The different experimental settings.

Finally, we would like to acknowledge the NumPy, Pandas and Scikit-Learn libraries [Oliphant, 2006, McKinney et al., 2010, Pedregosa et al., 2011].

B.1 Correcting Sparsified Predictions

In some cases the classification accuracy of a sparsified classifier is very poor even though the test AUC is good (e.g. accuracy 60% but AUC 0.80). The AUC depends on the relative ordering of predictions while classification accuracy depends on whether each prediction is above or below zero. This lead us to believe that the poor test classification accuracy was caused by a bad offset. Maybe we should predict $+1$ if points where above -0.01 instead of 0. In other words, it seemed that the sparsified classifiers skewed the bias term of the original classifier. To fix this we computed the bias term that yielded the largest classification accuracy on the training set. This can be done by sorting predictions and then trying every possible offset, one for each point, taking just $O(n \lg(n))$ time. For the airline dataset this typically improved the sparsified classifiers accuracy from 60% to 80%.

In this way we "corrected" the bias of all sparsified predictions. All test classification accuracies reported are corrected in this sense (including Figure 2).

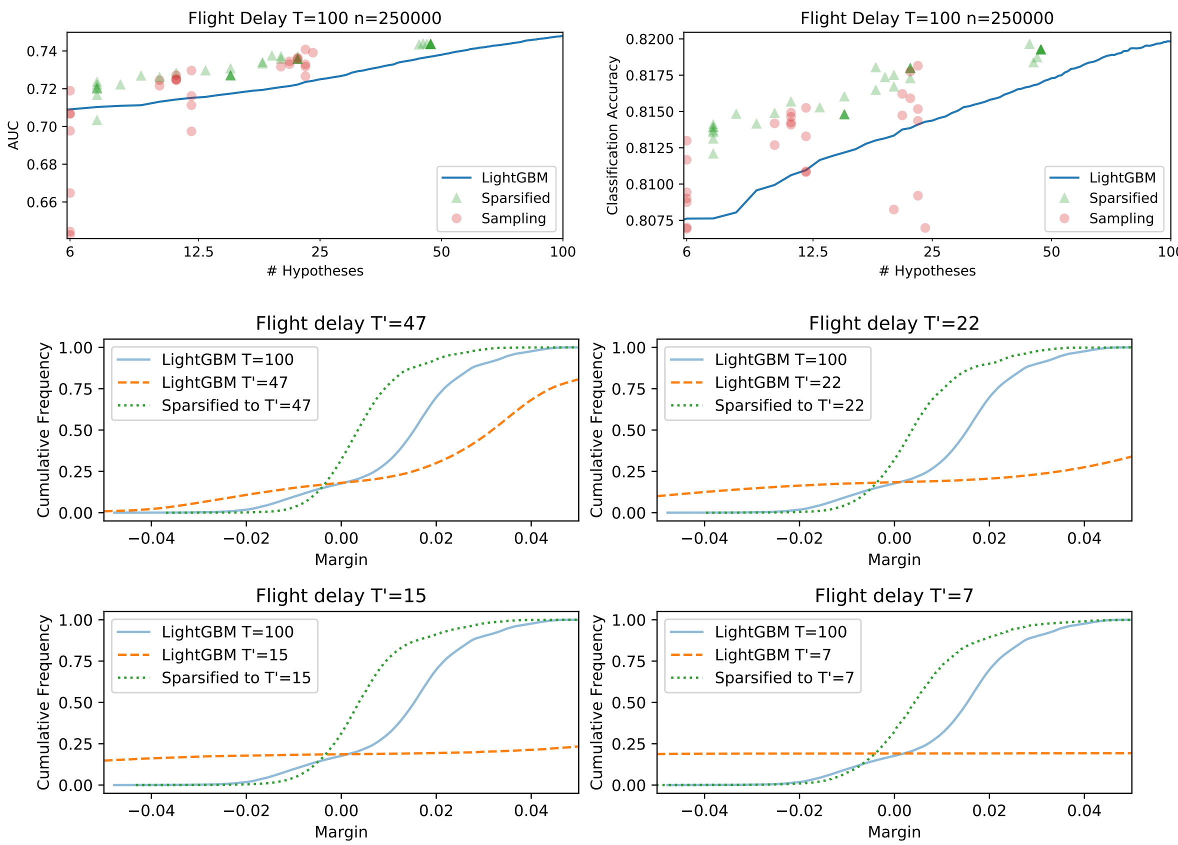

Figure 3: Similar to Figure 2 and Figure 1, but for Flight Delay with $T = 100$ and 250000 training and test points.

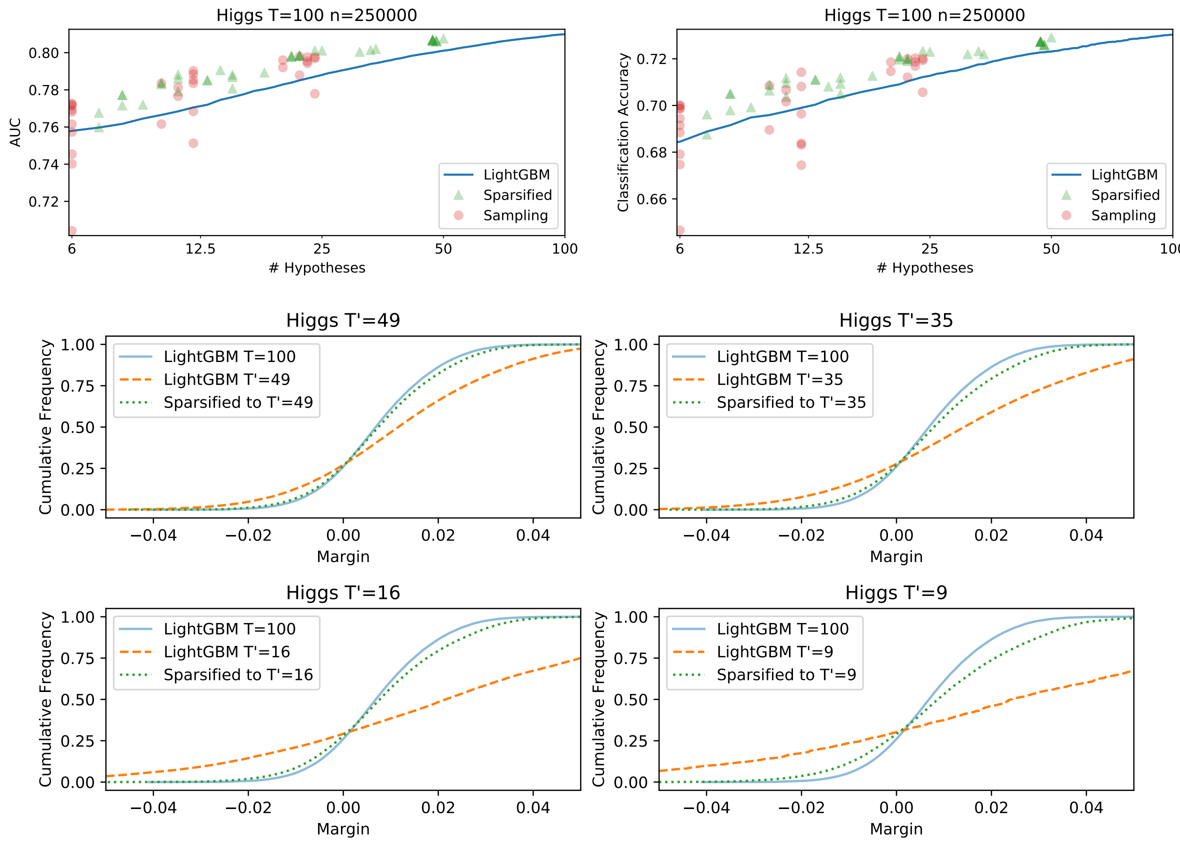

Figure 4: Similar to previous plot, but for Higgs with $T = 100$ hypotheses and 250000 training and test points.

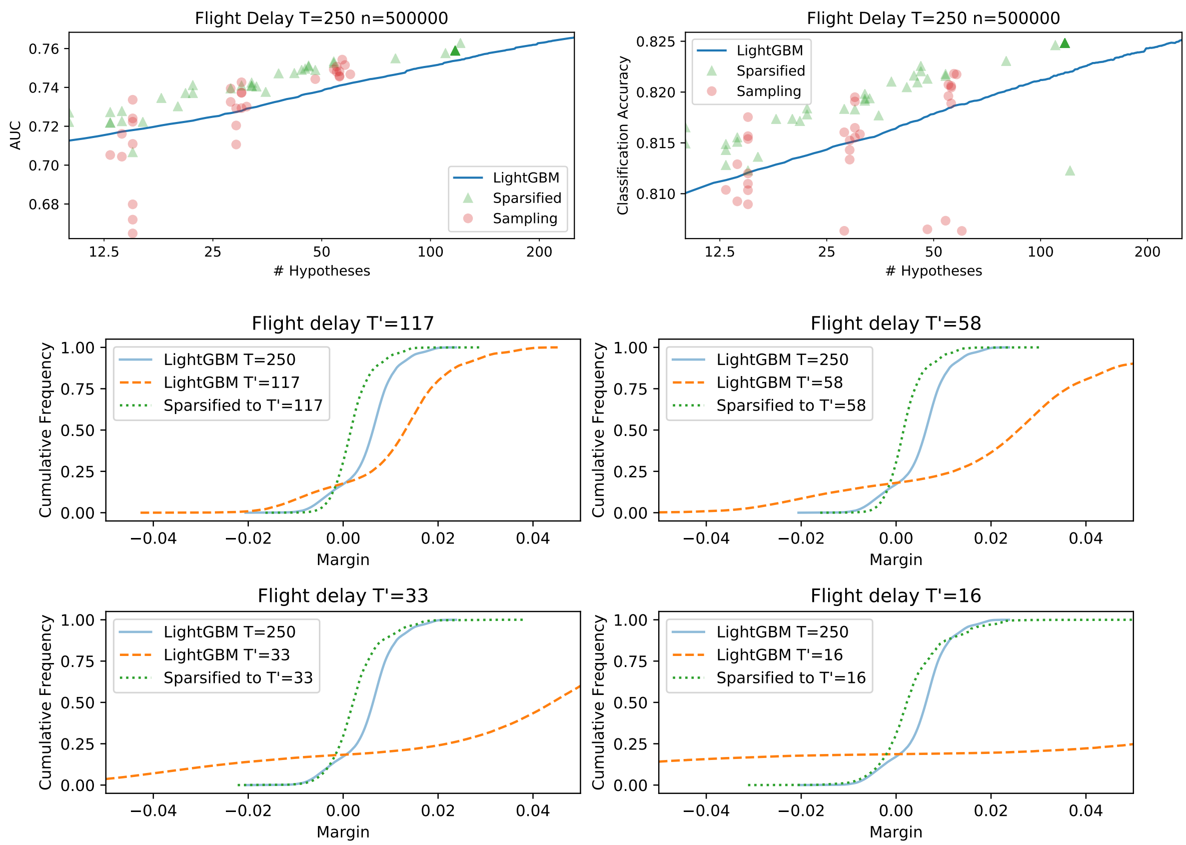

Figure 5: Similar to previous plot, but for Flight Delay with $T = 250$ hypotheses and 500000 training and test points.

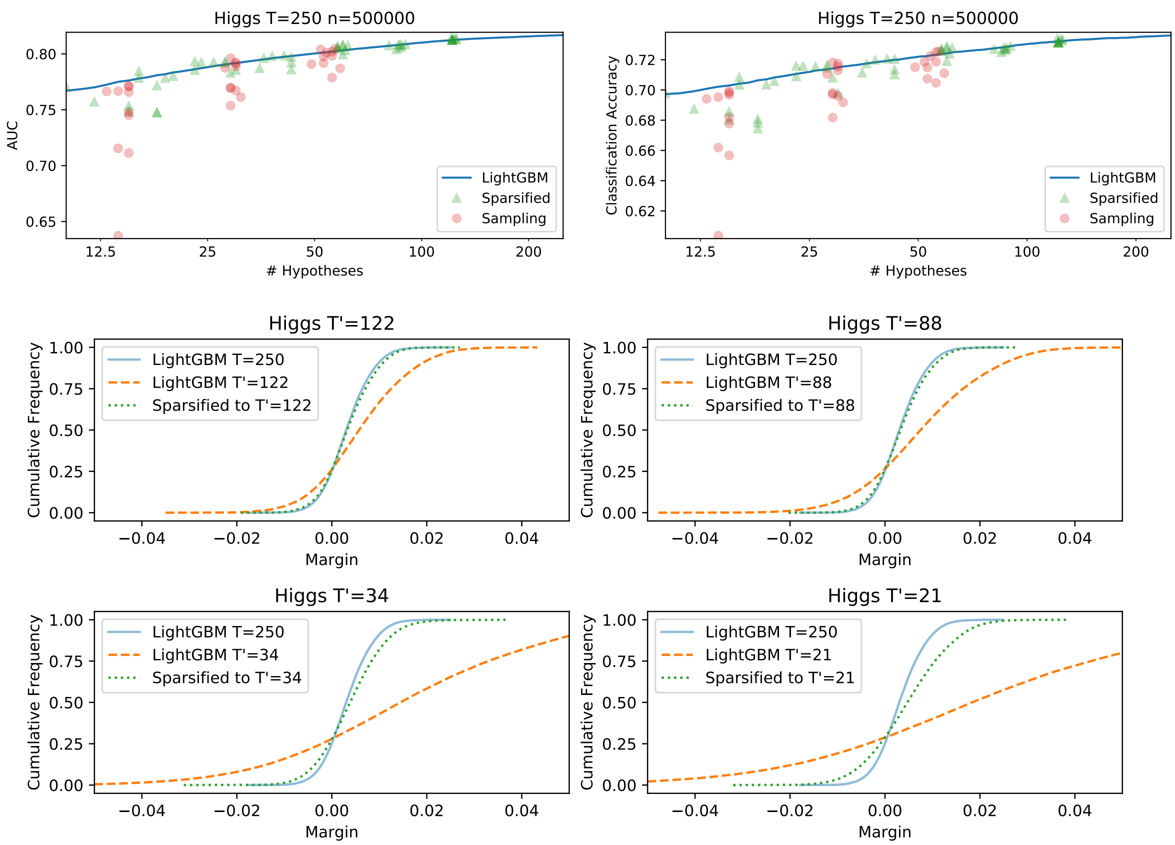

Figure 6: Similar to previous plot, but for Higgs with $T = 250$ hypotheses and 500000 training and test points.

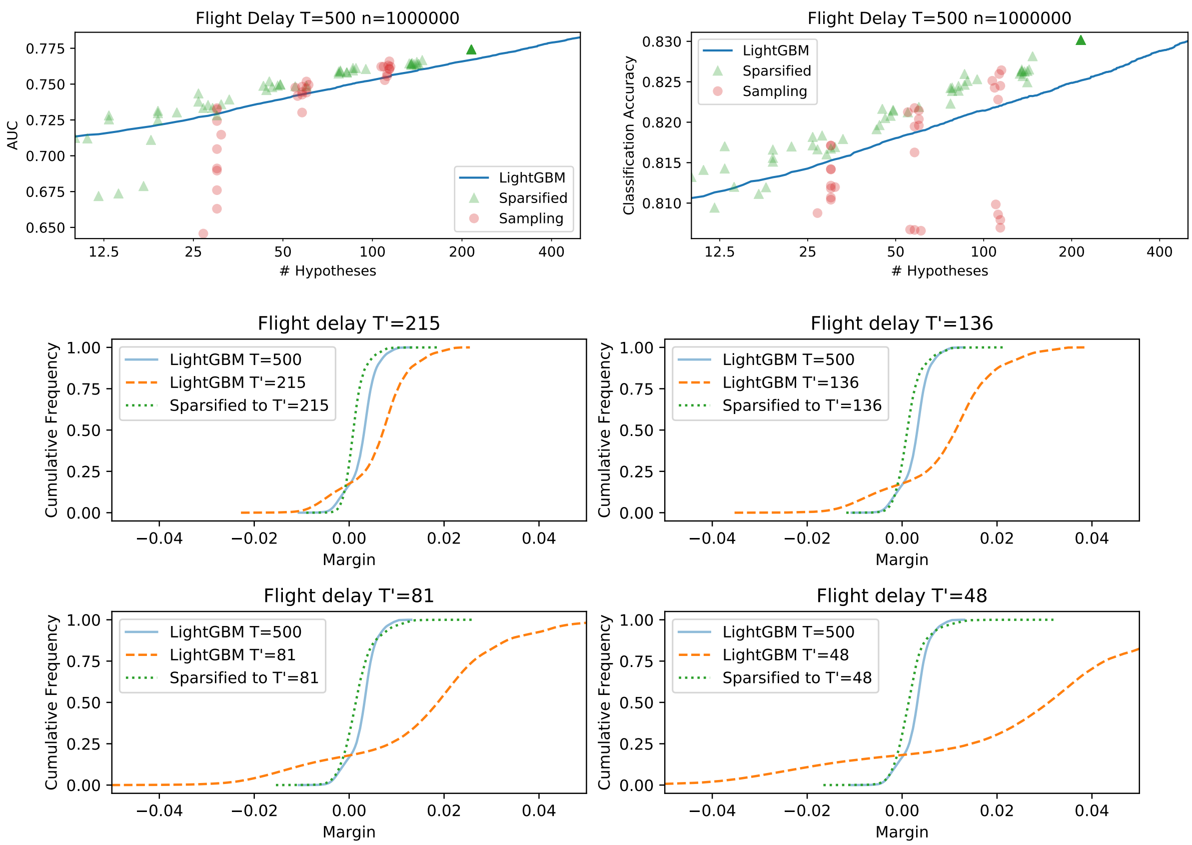

Figure 7: Similar to previous plot, but for Flight Delay with $T = 500$ hypotheses and 1000000 training and test points.

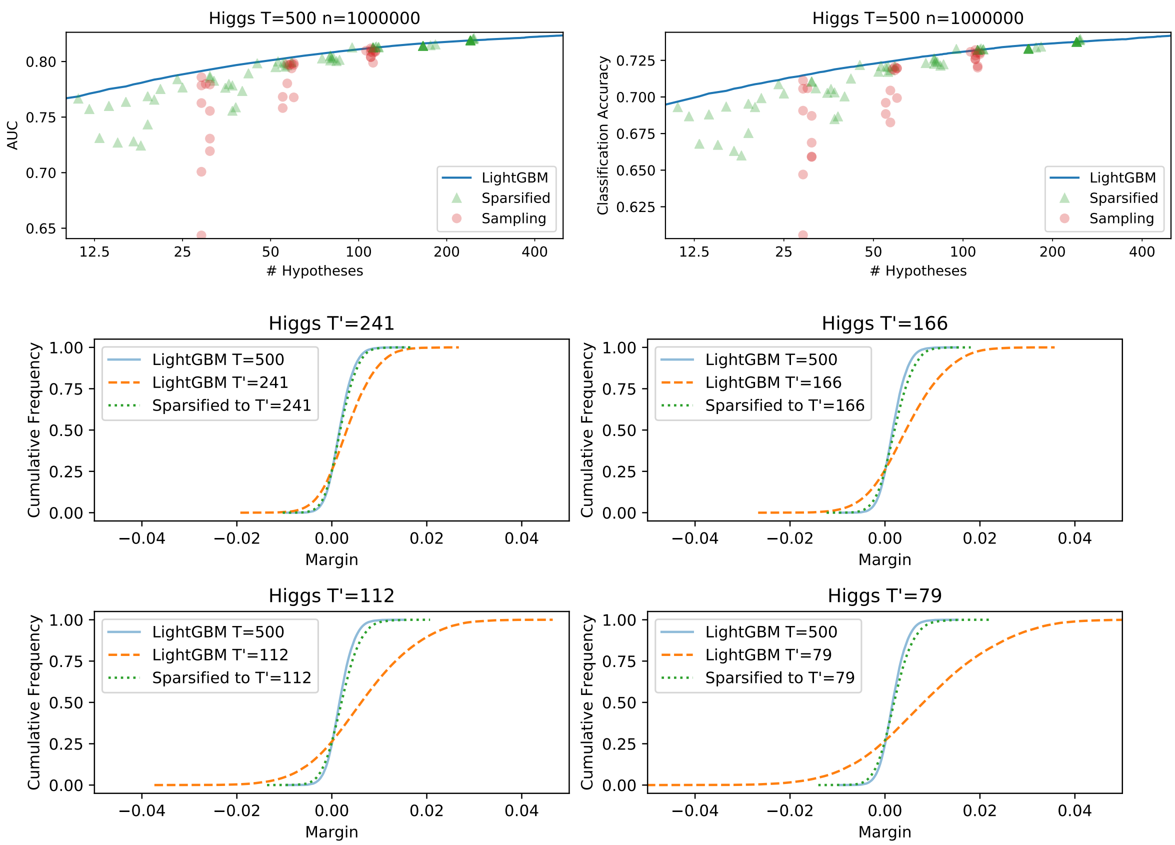

Figure 8: Similar to previous plot, but for Higgs with $T = 500$ hypotheses and 1000000 training and test points.

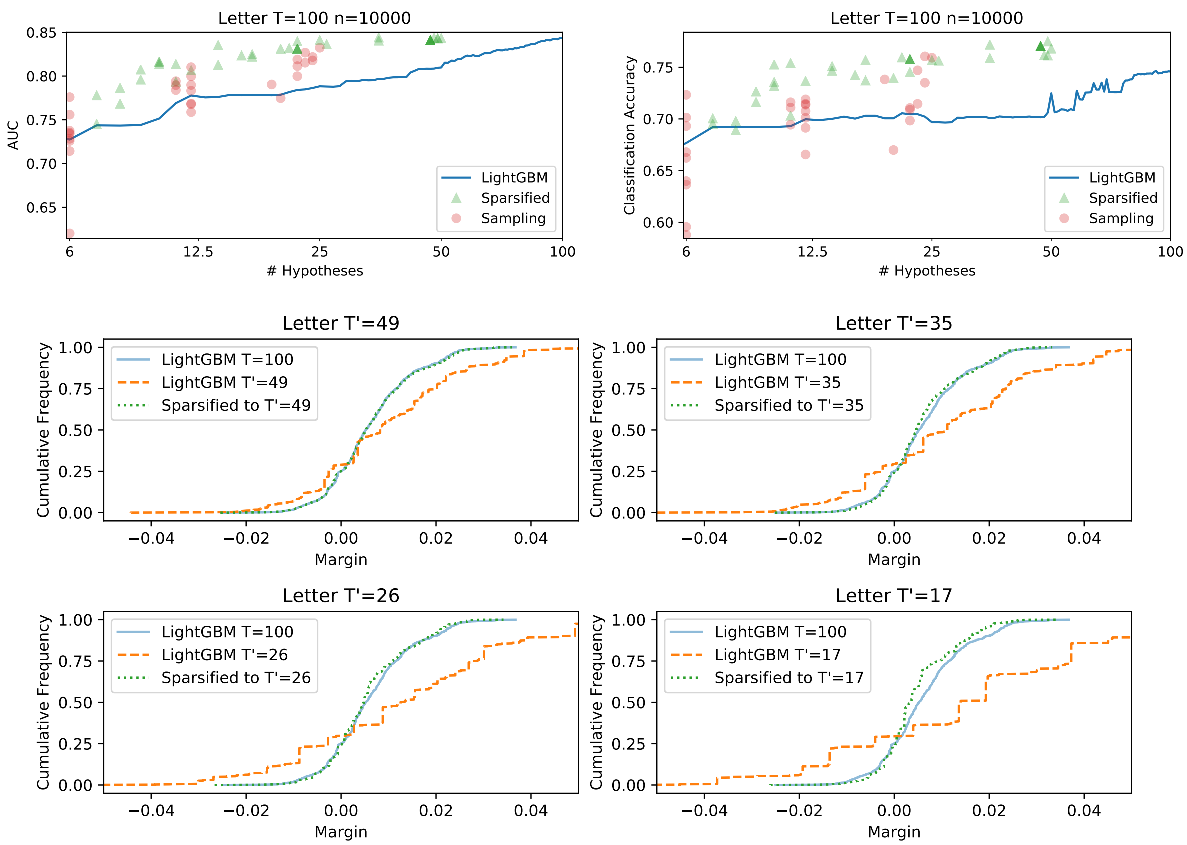

Figure 9: Similar to previous plot, but for Letter with $T = 100$ and $n = 10000$ training and test points. Furthermore, LightGBM used decision stumps (inspired by [Wang et al., 2008]), that is, we choose num_leaves $= 2$. Quite surprisingly the sparsified classifiers obtain a better classification accuracy than the original classifier. All sparsified classifiers predictions were corrected as described in Appendix B.1. Since the original classifier has better AUC than the sparsified ones, we believe this is caused by a poor bias for the original classifier.