Author Response ("Margins are Insufficient for Explaining Gradient Boosting")

We thank the reviewers for the time and expertise invested in these reviews.

Response to Reviewer 2. Addressing the reviewer's concerns, we wish to stress that while for Gradient Boosters (GB) the sigmoid function may be used to define the loss function and to transform the (raw) output prediction of a (learned) ensemble into a probability, the sigmoid function is not used when studying margin theory. We compute the margin of a an ensemble of base learners ($\{\alpha_i, h_i \mid h_i(x) \to [-1, 1]\}$) following classic margin theory (see Section 2). Loosely speaking, the margin of a point depends on the output of the voting classifier, and does not involve the sigmoid function. Formally, for data point x_i the margin is $\frac{\sum_{j=1}^{n} \alpha_j h_j(x_i)}{\sum_{j=1}^{n} |\alpha_j|}$. Note the final remark of Section 2 of how we transform the output of GB algorithms, e.g. LightGBM, to fit this framework. This is unencumbered by the loss function minimized in training or how sigmoid may be used to transform the output of the ensemble into to a probability distribution by setting $P(y = 1 \mid x) = \sigma(\sum_{j=1}^{n} \alpha_j h_j(x_i))$. We also note that as opposed to the margins, this probability is not scale-invariant as scaling the weights or adding copies of the base learners provide more extreme probabilities.

Albeit unconnected to margin theory, the sigmoid function cannot make large changes to the output if the input is only changed by a small amount. In fact small changes to the input to the sigmoid function makes even smaller changes to the output since the derivative of the sigmoid function is at most 1/4 (and that is only at zero and the norm of derivative goes to zero as $|x|$ goes to infinity very fast).

Regarding the reason the margins are reduced with the number of base regressors in GB, in short this is due to the fact that gradient boosters may decide to focus on only a small subset of data in each iteration Consider the margin of a fixed data point x, and assume for simplicity that all weights (α_j) are 1. Then the margin x is $\frac{1}{n} \sum_i h_i(x)$, where n is the number of base learners, which drops towards zero as n increases unless $h_i(x)$ remains large (that is, close to 1) for all $i = 1, \ldots, n$. Hence, if each round only focuses on a small subset of data points, meaning that the new base learner only gives non-negligible prediction on a small subset, then the margin of the remaining points decreases. If this happens in each round of boosting, the result is a smaller and smaller margin distribution, and this is what happens in the experiments shown in our paper. We will update our description of this argument in Section 3 of our paper to make it more clear (alse see the arguments presented in Section 4 under Potentially Much Better).

We thank the reviewer for the specific comments as well. We will certainly take care of them for the final version.

Response to Reviewer 1. As the reviewer suggests, the scope of our paper and theoretical contributions are more general than gradient boosted trees, as it applies to all voting classifiers irrespective of learning algorithm, including loss function optimized, and base learners. The paper investigates how the actual predictions of voting classifiers on the data may be very different than $\{-1, 1\}$, which is the standard assumption when proving margin bounds. Our theory, and margin theory in general, is orthogonal to specifics of the learning algorithm. We will think about ways of rephrasing our contributions to make it clearer that it is not only specific to gradient boosters with a concrete loss function. In practice, when conducting the numerical experiments, we chose to focus on comparing the classic AdaBoost algorithm where the base learner is restricted to map inputs to $\{-1, 1\}$ with Gradient Boosters that very often returns negligible predictions, as this exactly highlight and fits the phenomena we are investigating. We stress, however, that this is a special case, and the theoretical results we present are significantly broader.

For base learners, the same size means the same number of leaves (and no restriction on depth for both algorithms compared). We do not consider decision stumps as these are usually not used by GB in practice and lead to inferior performance (the default number of leaves for LightGBM is size 31, while XGBoost use a max depth default equal to 6). Furthermore, with decision stumps, the base classifiers are to weak for the phenomenon with small predictions to even occur. That is, with decision stumps, most predictions are very close to $\{-1, 1\}$. This is most likely due to the fact that decision stumps are incapable of "focusing" on a small subset of training points as discussed above. We will elaborate more on this in the final version and if space allows it, we will also include a histogram of predictions when using decision stumps.

In the supplemental material, submitted along with the paper, we included the same experiment on three more data sets to give 4 data sets of increasing size to analyze and demonstrate our new theoretical bound on. The observed behaviour was completely consistent. The mean validation error and standard deviation for the Forest Cover dataset example from the paper are (0.0298, 0.00037) for LightGBM and (0.0327, 0.00053) for AdaBoost. The standard deviation was so small that we chose to only show 3 runs on the plots. We will make sure to comment on this in a final version. For data sets included the in supp. material the mean, and standard deviation are as follows **Boone:** Lgb: mean=0.0574, std=0.00015, Ada: mean=0.0631, std=0.00068 **Higgs:** Lgb: mean=0.2530, std=0.00031, Ada: mean=0.2777, std=0.00009 **Diabetes:** Lgb: mean=0.2532, std=0.0255, Ada: mean=0.2692, std=0.0194 . for Boone and Higgs we used three runs and for Diabetes we used 10 (due to much smaller data size)