Can a Set of Calibrated Weak Learners Create a Single Calibrated Strong Learner?

Jason Park [*1] **Minseok Bae** [*1] **Yash Shah** [*1]

1. Introduction

Uncertainty calibration is crucial for creating trustworthy, reliable, and robust models. It aligns a model's prediction confidence with the actual likelihood of those predictions being correct (Guo et al., 2017). More formally, for a sample $x \in \mathcal{X}$, let $f(x) = \hat{p}$ denote the predicted probability by the classifier and $\hat{y}$ be the label assigned to x, where $\hat{y} = 1$ if $\hat{p} \geq 0.5$, else $\hat{y} = 0$. For perfect calibration, $\forall p \in [0, 1], \Pr[\hat{y} = y | \hat{p} = p] = p$.

The importance of model calibration escalates as we start applying models to safety-critical tasks such as autonomous driving (Chen et al., 2023), medicine and healthcare (Lin et al., 2022), face recognition (Kim et al., 2022), and time-series forecasting (Rasul et al., 2021). With deep neural networks deployed for a huge number of use cases (Ramesh et al., 2022; Stiennon et al., 2020; Dosovitskiy et al., 2021), researchers have addressed model calibration with novel algorithms for object detection (Pan et al., 2021), NLP tasks (Zhao et al., 2023), regression (Levi et al., 2020), multi-modal learning (Zhang et al., 2023), and so on. For traditional machine learning models, some of the classic post-hoc parametric and non-parametric calibration methods (Niculescu-Mizil & Caruana, 2005a) are still popular.

In this paper, we focus our attention on the AdaBoost ensemble algorithm (Freund & Schapire, 1995), known to be highly uncalibrated. We answer the question, "Can a set of calibrated weak learners create a single calibrated strong learner?" by making the following key contributions:

(1) We propose a novel algorithm that uses doubly-calibrated weak decision stumps to produce a single calibrated strong ensemble classifier following the AdaBoost learning algorithm.[2]

(2) We show an improvement for Thresholded Adaptive Calibration Error (TACE) over Platt Scaling, Isotonic Regression, and Beta Calibration on three datasets.

(3) We conduct extensive ablation studies to verify the utility and integrity of our proposed algorithm.

2. Related Works

We review existing parametric and non-parametric methods for calibrating traditional machine learning models, contextualizing our novel approach within the broader field:

Platt Scaling. A parametric approach (Platt, 1999) that fits a logistic curve to the predicted probabilities of a classifier $\hat{p}$ by learning parameters a and b to compute calibrated probabilities $\mathbb{P}[y = 1|\hat{p}] = \sigma(a\hat{p} + b)$ on a held-out set. The inspiration behind doing this is that uncalibrated model confidences typically exhibit a sigmoidal curve when visualized as a reliability diagram, which is a plot of expected sample accuracy as a function of model confidence (Niculescu-Mizil & Caruana, 2005b). Model predictions $\hat{p}$ are binned into M equally-spaced intervals and sample accuracy is calculated over predictions $\hat{y}$ that align with target labels y.

Isotonic Regression. A non-parametric method (Zadrozny & Elkan, 2001) that learns a piecewise constant function g over classifier predicted probabilities $\hat{p}$ by minimizing the squared loss $\sum_{i=1}^{n}(g(\hat{p}_i) - y_i)^2$. It is, however, known to overfit on small datasets since it is less constrained than Platt Scaling (Niculescu-Mizil & Caruana, 2005b).

Beta Calibration. Since Platt Scaling assumes Gaussian-distributed model predictions with infinite support, unreasonable for classifier outputs strictly in $[0, 1]$, Kull, Filho, and Flach (2017) use the beta distribution with finite support. This approach can also model the identity function, and is the current state-of-the-art post-hoc calibration method.

Calibrated Ensembles. (Kumar et al., 2022) show that a standard and a robust neural network when individually calibrated on in-domain held-out set and then ensembled by averaging the predictions mitigates accuracy tradeoffs under distribution shifts. In the traditional ML space, Niculescu-Mizil and Caruana (2005b) apply some of the above post-hoc calibration techniques to AdaBoost and analyze results.

3. Preliminaries

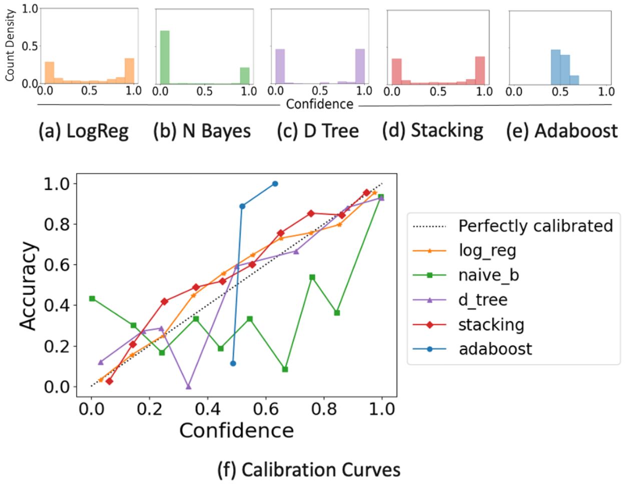
3.1. Problem Setup

Given a set $\mathcal{X}$ consisting of N samples, train a classifier f to predict a binary label $y \in \{0, 1\}$ for each $x \in \mathcal{X}$.

[*] Equal contribution, order decided by coin flip. [1] Stanford University. Correspondence to: Jason, Minseok, Yash <{jpark26,minseok,ynshah}@stanford.edu>.

[2] Code available at https://github.com/ynshah3/CS229Calibration

3.2. Problem Motivation



(a) LogReg (b) N Bayes (c) D Tree (d) Stacking (e) Adaboost

(f) Calibration Curves

Figure 1: [a-e] show confidence histograms (count density vs model predicted probabilities) and [f] shows the reliability diagram for uncalibrated models evaluated on the test set of landsat dataset.

We train five different models on the landsat dataset (see section 4.2): Logistic Regression, Naive Bayes, Decision Tree, a stacked-ensemble of these 3 models, and AdaBoost. Model parameters are mentioned in section 4.1. Figure 1 shows that Logistic Regression, Naive Bayes, Decision Tree, and the Stacked-ensemble produce confidences such that they are very confident on a large fraction of the unseen input samples (seen as peaks at around 0 and 1).

AdaBoost, on the other hand, is highly underconfident; its predicted probabilities for each unseen sample are clustered around 0.5. Moreover, it has a calibration curve that is almost vertical, as opposed to the other four models having curves close to the $y = x$ line implying perfect calibration. Since AdaBoost is highly uncalibrated, several post-hoc calibration techniques like Platt Scaling, Isotonic Regression, and Beta Calibration have been applied by previous works (performance results in section 5.3). These methods, while calibrating the model to an extent, do not produce perfect calibration and are applied post training. A natural question to ask is: given the idea behind Kumar, Ma, Liang, and Raghunathan (2022) and the fact that a stacked-ensemble is implictly calibrated when it is composed of individually calibrated classifiers, can we apply the same learning-based idea to AdaBoost? We answer this in the following section by proposing a novel algorithm that achieves better calibration errors on multiple datasets.

4. Methods

4.1. Models and Parameters

Logistic Regression: Solves the quasi-Newton Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm (Liu & Nocedal, 1989) by computing estimates of the inverse Hessian matrix for optimization by keeping a history of the past m gradient evaluations to save memory. This is computationally efficient since inverting the actual Hessian for computing the approximate Newton direction for optimization ($p = -H^{-1}g$) is expensive. The inverse Hessian itself is approximated using rank-one updates gradually, starting with the identity matrix.

Naive Bayes: A Multinomial Naive Bayes, suitable for frequency count data and often used in text classification.

Decision Tree: Configured with a max depth of 10; uses Gini impurity for node splitting.

Stacked Ensemble: Combines Logistic Regression, Naive Bayes, and Decision Tree models into an ensemble. The outputs of these classifiers are fed into a Logistic regression meta-classifier trained via 5-fold cross-validation.

AdaBoost: Adopts 200 decision stumps with max-depth 1, sequentially fitted with re-weighting of samples to focus on challenging instances. We use the real version of the Stage-wise Additive Modeling using a Multi-class Exponential loss function (SAMME.R) (Zhu et al., 2006) for its quick convergence, detailed in section 4.3.

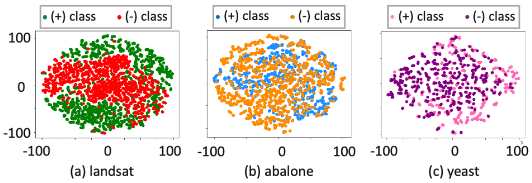
4.2. Datasets

NAME	SAMPLES	FEATURES	$(-)$ CLASS	$(+)$ CLASS
LANDSAT	6435	36	$\{1, 7\}$	$\{2, \ldots, 6\}$
ABALONE	4177	8	$\{1, ..., 9\}$	$\{10, \ldots, 29\}$
YEAST	1484	8	$\{1, 3\}$	$\{2, 4, \ldots, 10\}$

We use the following three datasets from the UCI Machine Learning Repository (Kelly et al.) for binary classification:

- **landsat**: predict soil type from multi-spectral values of pixels in 3x3 tiles of a satellite image.
- **abalone**: predict age of abalone given sex, length, weight, rings, diameter, etc.
- **yeast**: predict cellular localization sites of proteins given numeric measurements and signals.

We do not perform any other pre-processing on these datasets. The plots presented below are created using T-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008), an unsupervised, non-linear, dimensionality-reduction technique which minimizes the KL divergence between joint probabilities of low-dimensional embeddings and high-dimensional data.



(a) landsat (b) abalone (c) yeast

Algorithm 1 Double Calibration for AdaBoost

```
Require: $s$: Number of weak learners
Require: $\gamma \in (0, 1)$: Fraction of input samples to be reserved for calibration. Rest will be used for training
Require: $k$: Hyperparameter for $k$-fold cross validation
Require: $(\mathcal{X}, \mathcal{Y})$: set of input samples and labels
for weak learner $e_i \in \{e_1, e_2, \dots, e_s\}$ do
    $(\mathcal{X}_T, \mathcal{Y}_T), (\mathcal{X}_C, \mathcal{Y}_C) \leftarrow$ split$((\mathcal{X}, \mathcal{Y}), \gamma)$ {reserve $\gamma$ fraction of the samples randomly for calibration}
    $w \leftarrow$ sample weights from previous step; uniform if first step
    $e'_i \leftarrow$ using a $k$-fold cross validation procedure, fit $e_i$ on $(k-1)$ folds of weighted $(\mathcal{X}_T, \mathcal{Y}_T)$, calibrate $e_i$ on the $k^{\text{th}}$
        fold with the same weights $w$, and return predicted probabilities as average over calibrated $e_i$'s from each fold
    $e''_i \leftarrow$ calibrate $e'_i$ on $(\mathcal{X}_C, \mathcal{Y}_C)$ without any sample weighting
    Use $e''_i$ as the calibrated weak learner for current step to make predictions on unseen data
    Use predicted probabilities from $e'_i$ to update weights for the next step
end for
```

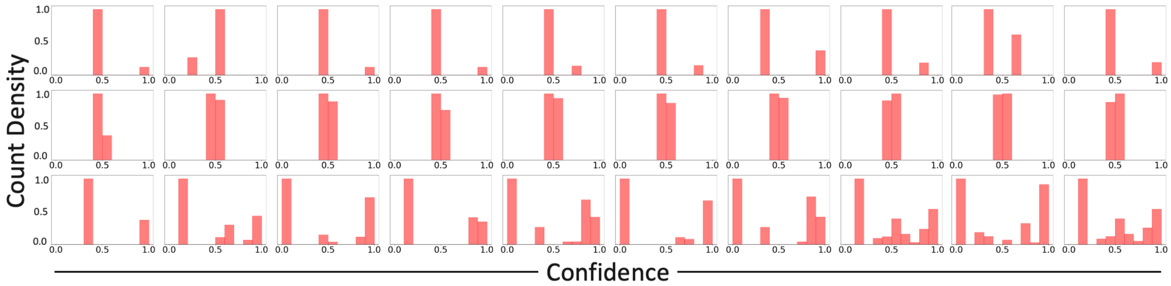


Figure 2: Confidence histograms for 10 of the 200 weak learners. First row corresponds to the uncalibrated weak learners e_i, second to singly-calibrated e'_i, and third to doubly-calibrated e''_i.

4.3. Algorithm

AdaBoost, optimized using the SAMME.R algorithm, works as follows: at each step $i \in [s]$, a weak learner e_i is fitted on w weighted training data. Then, weighted class probability estimates $\hat{p_j}$ are obtained on this data. For a decision stump fitted on N training samples, a sample j gets a score of $\hat{p_j} = k/N$, where k is the number of positive samples in the leaf node that j gets mapped to. $\hat{p_j}$ is used to update sample weights w for the next step. Final predicted probabilites on unseen x become $\sum_{i=1}^{s} e_i(x)$.

We make the following modifications: Before starting the algorithm, we reserve γ fraction of the training data for calibration. Then, at each step i of the algorithm, instead of using $\hat{p_j}$ from weak learner e_i directly, we first use a k-fold cross validation procedure to train e_i on $(k-1)$ folds of w weighted samples, then calibrate it on the k^{th} fold using same weights w. This means that the calibration method optimizes weighted loss $w_j \ell_j$ for each sample j instead of just ℓ_j, which it would have done without sample weighting. We call this singly calibrated model e'_i, the average over k choices; probabilities $\hat{p_j}$ now come from e'_i and are used to update the weights w for the next step. We also save e''_i which is obtained after calibrating e'_i on the γ-held-out set. Final predicted probabilities, hence, become $\sum_{i=1}^{s} e''_i(x)$.

4.4. Why Double Calibration?

Since uncalibrated e_i (a decision stump of depth 1) receives weighted samples as input, it is trained to perform better on those with higher weights. As a result, we see in the first row of figure 2 that one of the child nodes contains highly weighted samples that e_i performs well on—resulting in a smaller confident peak closer to 0 or 1—while the other child node contains the rest of the samples (and thus, a similar number of positive and negative samples), producing a larger underconfident peak near 0.5.

In obtaining e'_i, we calibrate e_i for the first time with weighted samples. This step puts greater emphasis on calibrating probabilities for samples with higher weights, akin to them being "seen multiple times" during training. This effectively mitigates the effect that sample weighting has in training the weak learners (which resulted in skewed predictions), bringing confidence peaks closer to 0.5.

Finally, since e'_i only undoes the sample weighting effect, for true calibration, we must calibrate once again, without sample weights. This yields e''_i, which in figure 2 is shown to produce a spread of predicted probabilities expected from a calibrated classifier. The doubly-calibrated e''_i can thus attribute correct confidences to unseen test samples, unaffected by the weights used during training.

Table 1: Evaluation on the test set. First five models are uncalibrated (trained on 100% of the data). Train ACC and TACE for Platt, Isotonic, and Beta are computed after training AdaBoost on 80% of the data and before performing the respective calibration methods. Among variants of AdaBoost, best results are marked in **bold**.

DATASET	METRIC	LOGREG	NBAYES	DTREE	STACKING	ADABOOST	PLATT	ISOTONIC	BETA	OURS
LANDSAT	TRAIN ACC (↑)	0.886	0.640	0.964	0.947	**0.930**	0.920	0.920	0.920	0.895
	TEST ACC (↑)	0.870	0.648	0.892	0.898	0.900	0.900	0.897	**0.902**	0.882
	TRAIN TACE (↓)	0.021	0.266	0.0	0.061	0.397	0.396	0.396	0.396	**0.036**
	TEST TACE (↓)	0.031	0.259	0.063	0.044	0.365	0.047	0.043	0.044	**0.032**
ABALONE	TRAIN ACC (↑)	0.769	0.687	0.900	0.818	**0.800**	0.798	0.798	0.798	0.754
	TEST ACC (↑)	0.785	0.698	0.725	0.792	**0.788**	0.783	0.795	0.788	0.755
	TRAIN TACE (↓)	0.031	0.075	0.0	0.049	0.282	0.283	0.283	0.283	**0.043**
	TEST TACE (↓)	0.034	0.072	0.155	0.035	0.276	0.054	0.051	0.048	**0.048**
YEAST	TRAIN ACC (↑)	0.676	0.533	0.877	0.820	**0.785**	0.783	0.783	0.783	0.724
	TEST ACC (↑)	0.663	0.505	0.707	0.707	**0.734**	0.714	0.720	0.714	0.731
	TRAIN TACE (↓)	0.069	0.151	0.0	0.134	0.277	0.273	0.273	0.273	**0.045**
	TEST TACE (↓)	0.084	0.142	0.161	0.073	0.224	0.066	0.063	0.061	**0.056**

5. Experiments and Discussion

5.1. Evaluation Metrics

Our experimental results are evaluated on two metrics: Accuracy (ACC) and Thresholded-Adaptive Calibration Error (TACE). TACE is computed with a threshold of $\epsilon = 0.01$, considering only predictions that exceed this threshold to avoid infinitesimal confidences. Furthermore, we utilize a class-conditioned adaptive binning approach with 10 bins per class, where the predictions are first split based on true class and then bin intervals are set to evenly distribute predictions across bins. The TACE is then defined as:

$$TACE = \frac{1}{KR}\sum_{k=1}^{K}\sum_{r=1}^{R}|\text{acc}(r,k) - \text{conf}(r,k)|,$$

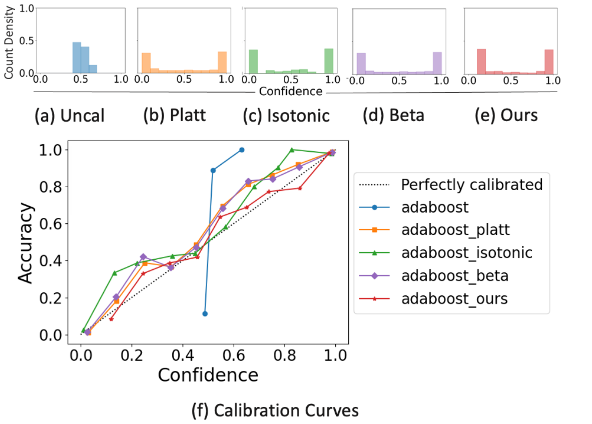
where $K = 2$ is the number of classes, $R = 10$ is the number of bins, and $\text{acc}(r,k)$ and $\text{conf}(r,k)$ correspond to the accuracy and confidence of bin r for class k respectively. TACE is chosen over Expected Calibration Error (ECE) due to the latter's issues in bias-variance tradeoff and pathologies in static binning schemes (Nixon et al., 2020).

5.2. Hyperparameters

Hyperparameters are determined by using a 5-fold cross-validation procedure on the training set to avoid overfitting, and keeping the test set unseen: $\gamma = 0.05$, $k = 5$ for landsat and yeast, and $\gamma = 0.04$, $k = 5$ for abalone. The order of double calibration is Platt Scaling then Beta calibration for landsat, and Platt Scaling followed by Platt Scaling again for abalone and yeast. Why we use these specific hyperparameters is detailed in section 5.4. To make comparisons fair, we reserve 20% of the data for calibration for post-training calibration methods (since our model uses 5-fold CV, which trains the model on 80% of the data every time).

5.3. Results

All figures are plotted for the landsat dataset, but show similar trends for other datasets as well. The results, as shown in table 1, demonstrate our model's competitive performance. Notably, our approach achieves lower TACE than post-training calibration methods applied to AdaBoost across all three datasets, indicating a significant improvement in calibration. Furthermore, while our model exhibits slightly lower accuracy scores when compared to other variants of AdaBoost, it generally outperforms other traditional machine learning models. This is especially apparent in a more complex dataset like yeast. Logistic Regression and Stacking have a decent TACE even when uncalibrated, but suffer from either lower accuracy or a very long runtime. Therefore, we see that our model allows us to achieve the best of both worlds, excelling in both accuracy and calibration. Qualitatively, figure 3 shows that our model's calibration curve is observably closest to the ideal and has learned to not be highly underconfident as it was when uncalibrated.



(a) Uncal (b) Platt (c) Isotonic (d) Beta (e) Ours

(f) Calibration Curves

Figure 3: [a-e] show confidence histograms and [f] shows the reliability diagram for AdaBoost with different calibrations applied.

It is worth noting that while post-training calibration techniques are fast and efficient, they often lack the flexibility of a training-based approach like ours. Although our model incurs higher computational demands during the training phase, this cost is a single-time incurrence and is recompensed through the possibility of balancing the accuracy-calibration error tradeoff for specific datasets. Thus, our model not only achieves better calibration but also maintains high accuracy, illustrating the benefits of our training-based calibration approach in diverse ML scenarios.

5.4. Ablation Studies

5.4.1. Singly-calibrated weak learners

Could we have achieved the same competitive results by only calibrating weak learners once using k-fold CV with non-weighted samples?

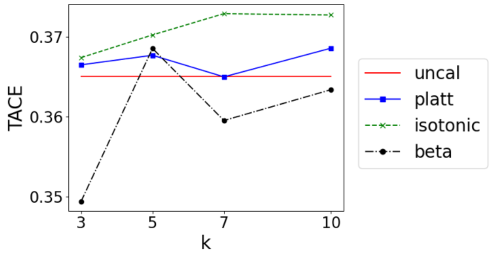


Figure 4: Performance of our AdaBoost with singly-calibrated weak learners as k is varied on landsat dataset.

As shown in Figure 5, the models are still highly uncalibrated. The general trend for TACE, though noisy, is that it tends to get worse with large k's (smaller folds) as we reserve lesser portions of the data for calibration, making these sets less representative of the entire distribution.

5.4.2. k and γ vs. TACE

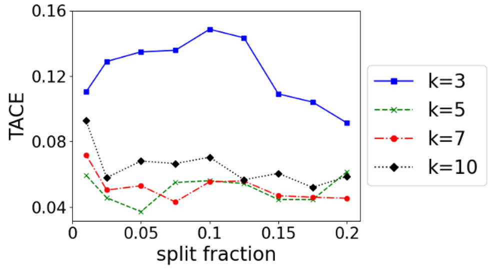


Figure 5: Plot of how changing k and γ affect TACE on landsat dataset using the double calibration order platt → beta.

Our model requires a trade-off between k and γ, as both are needed for true calibration, explained in section 4.4. The weight-undoing (first) calibration step risks overfitting with large folds (see $k = 3$) and underfitting with small, non-representative folds (see $k = 10$). In general, TACE tends to decrease as γ increases, albeit with noise, by having more data for the second calibration step.

5.4.3. Order of double calibration

As seen in Figure 6, the order in which different calibration techniques are applied affects the error. Since the task of the first calibration method is to undo the effect of weighted samples, Isotonic Regression — a non-parametric, monotonic, piecewise function minimizing squared loss — is not able to model the complexities of the distribution of these weighted sample confidences as effectively as Platt Scaling is, which fits a sigmoid curve (which is also the distribution of model confidences) by minimizing cross-entropy.

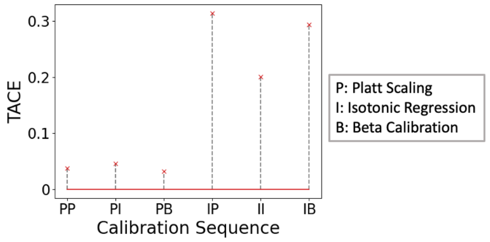


Figure 6: Performance of our AdaBoost with varied ordering of the two calibration methods for doubly-calibrated weak learners.

5.4.4. Number of weak learners vs. ACC and TACE

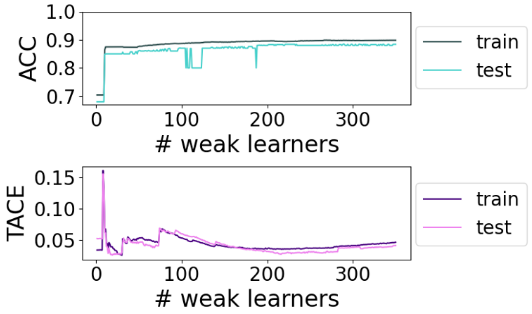


Figure 7: Performance of our AdaBoost with varied number of weak learners on landsat dataset.

Figure 7 demonstrates that both accuracy and TACE scores become better as we increase the number of weak learners in our AdaBoost algorithm. This is because more classifiers can contribute their data modellings to the ensemble. There is, however, a threshold (at around 250) when TACE starts to increase again, which is when variance in the learners starts to dominate. Finding a good balance helps achieve optimal metric scores.

6. Conclusion and Future Work

Our study illustrates that calibrated weak learners can form a well-calibrated AdaBoost model, potentially surpassing traditional post-hoc methods such as Platt scaling in performance. The "double calibration" strategy introduced here—first to counteract AdaBoost's sample weight bias, then to calibrate the model—offers performance benefits and enhanced adaptability during the learning phase. Despite these advantages, the approach incurs greater computational costs during training. Future efforts will focus on analyzing and optimizing the computational efficiency compared to post-hoc calibration techniques. Additionally, we aim to extend our research to multi-class classification tasks to broaden the utility of our findings. This dual approach will refine the calibration strategy, striving for a balance between accuracy and computational practicality.

7. Team Contributions

Jason Park. Carried out implementation for AdaBoost with LogReg decision trees in an attempt for inherent model-based calibration. Also worked on some of the ablation studies, the experiments and discussion section of the report, and overall editing.

Minseok Bae. Explored existing learning-based calibration methods and implemented LogReg decision trees. Worked on methods, most of the ablation studies, and conclusion sections on this report.

Yash Shah. Worked on coming up with the double-calibration algorithm and implementing that in code. Carried out some of the ablation studies, and worked on the introduction, related works, and algorithm sections on this report.

References

Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., and Li, H. End-to-end autonomous driving: Challenges and frontiers. *arXiv*, 2306.16927, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Freund, Y. and Schapire, R. E. A desicion-theoretic generalization of on-line learning and an application to boosting. In Vitányi, P. (ed.), *Computational Learning Theory*, pp. 23–37, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. ISBN 978-3-540-49195-8.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Kelly, M., Longjohn, R., and Nottingham, K. The uci machine learning repository. URL https://archive.ics.uci.edu.

Kim, M., Jain, A. K., and Liu, X. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18750–18759, June 2022.

Kull, M., Filho, T. S., and Flach, P. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 623–631. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/kull17a.html.

Kumar, A., Ma, T., Liang, P., and Raghunathan, A. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=S2EuSP8sqgc.

Levi, D., Gispan, L., Giladi, N., and Fetaya, E. Evaluating and calibrating uncertainty prediction in regression tasks, 2020. URL https://openreview.net/forum?id=ryg8wpEtvB.

Lin, Y., Yao, H., Li, Z., Zheng, G., and Li, X. Calibrating label distribution for class-imbalanced barely-supervised knee segmentation, 2022. URL https://arxiv.org/abs/2205.03644.

Liu, D. and Nocedal, J. On the limited memory bfgs method for large scale optimization. In *Mathematical Programming 45*, pp. 503–528, 1989. URL https://doi.org/10.1007/BF01589116.

Niculescu-Mizil, A. and Caruana, R. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005a. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102430. URL https://doi.org/10.1145/1102351.1102430.

Niculescu-Mizil, A. and Caruana, R. Obtaining calibrated probabilities from boosting. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pp. 413–420, Arlington, Virginia, USA, 2005b. AUAI Press. ISBN 0974903914.

Nixon, J., Dusenberry, M., Jerfel, G., Nguyen, T., Liu, J., Zhang, L., and Tran, D. Measuring calibration in deep learning, 2020.

Pan, T.-Y., Zhang, C., Li, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B., and Chao, W.-L. On model calibration for long-tailed object detection and instance segmentation. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2529–2542. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/14ad095ecc1c3e1b87f3c522836e9158-Paper.pdf.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61-74, 06 1999.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022.

Rasul, K. Uncertainty Metrics, November 2020. URL https://github.com/kashif/uncertainty-metrics.

Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U. M., and Vollgraf, R. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=WiGQBFuVRv.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 609–616, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Zhang, H. M. Q., Zhang, C., Wu, B., Fu, H., Zhou, J. T., and Hu, Q. Calibrating multimodal learning, 2023.

Zhao, T., Wei, M., Preston, J. S., and Poon, H. Llm calibration and automatic hallucination detection via pareto optimal self-supervision, 2023.

Zhu, J., Rosset, S., Zou, H., and Hastie, T. Multi-class adaboost. *Statistics and its interface*, 2, 02 2006. doi: 10.4310/SII.2009.v2.n3.a8.