Calibration for Anomaly Detection

Adrian Benton
abenton10@bloomberg.net
Bloomberg LP
New York, NY

ABSTRACT

Recent work on model calibration found that a simple variant of Platt scaling, temperature scaling, is effective at calibrating modern neural networks across an array of classification tasks. However, when negative examples overwhelm the dataset, classifiers will often be biased to producing well-calibrated predictions for negative examples, but have trouble producing well-calibrated predictions for true anomalies. A well-calibrated model – one whose scores accurately reflect the true probability of anomaly likelihood – is an invaluable tool for decision makers.

We present a short review of standard model calibration methods and propose an extension of Platt scaling, Charcoal Grill Scaling, to account for unevenly calibrated score distributions which arise from models trained to predict unlikely events. We compare CGS to existing calibration methods when calibrating models trained on balanced class distributions, as well as severely imbalanced class distribution in credit card fraud prediction.

CCS CONCEPTS

• **Computing methodologies** → Machine learning.

KEYWORDS

model calibration, class imbalance

1 INTRODUCTION

Anomalies are often associated with great risk: missing an instance of credit card fraud allows for more fraudulent withdrawals, not anticipating increased volatility in a security could lead to a lost investment. Statistical models are routinely used to estimate the probability of anomalies, but ultimately a human must act on the model output. A well-calibrated model – one whose scores accurately reflect the true probability of anomaly likelihood – is an invaluable tool for decision makers.

Recent work on model calibration found that a simple variant of Platt scaling, temperature scaling, is effective at calibrating modern neural networks across an array of classification tasks. However, this work only considers tasks with balanced classes, which is not appropriate for anomaly detection tasks. When negative examples overwhelm the dataset, classifiers will often be biased to producing well-calibrated predictions for negative examples, but have trouble producing well-calibrated predictions for true anomalies.

To remedy this issue, we propose an extension of Platt scaling, Charcoal Grill Scaling (CGS), that uses a mixture of Gaussians laying in logit space to govern the degree by which each score is scaled. CGS learns a smooth calibration function, unlike isotonic regression, but allows more flexibility to fit unevenly calibrated score distributions. We compare CGS to standard calibration methods when calibrating models trained on balanced class distributions (the Stocknet, MNIST, CIFAR-100, and SVHN datasets), as well as severely imbalanced distributions (credit card fraud prediction).

Section 2 briefly describes the calibration problem for binary classifiers and how calibration model quality is typically evaluated. In Section 3, we review existing calibration models and their effect on a synthetic dataset of miscalibrated scores. In Section 4, we present a novel extension of Platt scaling that allows for more flexible, smooth calibration functions. Section 5 describes the datasets we evaluate calibration models on. Section 6 describes the performance of each of those models. Finally, Section 7 reflects on which situations may be the most appropriate for CGS vs. existing methods.

2 MODEL CALIBRATION

Suppose you have trained a probabilistic binary classifier that outputs a score in $[0, 1]$ for every given example. The question is: can a user treat these scores these probabilities? For examples that are scored with 0.9, are 90% of these examples truly positive? If the probabilities assigned by our classifier can be trusted, then the classifier is said to be *well-calibrated*. Well-calibrated classifiers are especially desirable in cases where a person will use the classifier output to make consequential decisions, such as the financial or medical domains. The output of poorly calibrated classifiers can still be very useful as a feature in downstream models, or to rank examples, but it should not be interpreted as a probability.

Calibration of Neural Networks. Niculescu-Mizil and Caruana [8] showed that neural networks with softmax output layers, that explicitly predicted probabilities, were well-calibrated. As neural networks became more baroque and their training more arcane, this property no longer holds. In fact, Guo et al. [2] showed that although modern deep neural networks are much more accurate compared to older architectures, they are also far too confident in their predictions, producing probabilities that are poorly calibrated. Based on evaluation over a wide range of datasets, they recommend using a simple form of Platt scaling, *temperature scaling*, to calibrate model outputs. Although temperature scaling works well on standard evaluation sets, especially shared image recognition tasks, it may be less effective when classes are highly imbalanced or are very cost-sensitive.

In general, a calibration function adjusts model outputs to better match the actual probabilities that the event will occur. In this work we assume that the model logits are given to us and that the calibration function adjusts these logits to improve model calibration. Formally, we want to learn a calibration function $f : \mathbb{R}^{|Y|} \rightarrow \mathbb{R}^{|Y|}$ that adjusts the logits output by a classifier, where $|Y|$ are the number of classes in our data[1].

[1]For the binary case, $f : \mathbb{R} \rightarrow \mathbb{R}$ only needs to adjust a single score, for the positive class.

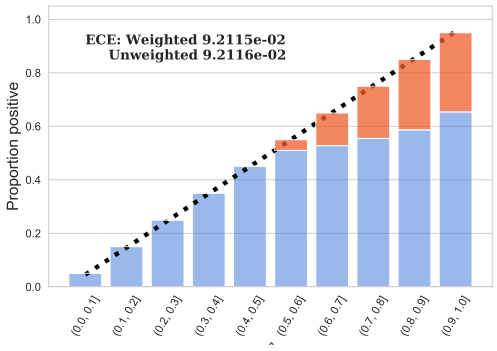

Figure 1: Example of a reliability diagram for synthetically generated scores.

2.1 Evaluating Calibration

Equation (1) gives the formal definition of a well-calibrated classifier.

$$\forall p \in [0,1], y \in \{1,\ldots,|Y|\}; \mathbb{P}(Y = y|\hat{p}_y = p) = p \tag{1}$$

where Y is a random variable of the example class, y is a fixed class, $\hat{p}_y$ is the predicted probability that the example will belong to class y, and p is an arbitrary probability. In practice, we can never verify whether this property holds since that would entail applying our model to its entire domain (infinite size or at least far too large to practically enumerate). In practice, calibration is evaluated by looking at small ranges of scores (bins) and estimating the above probability within each bin. One way to visualize the discrepancy between a well-calibrated classifier and one's own model is through a *reliability diagram*.

Reliability Diagrams. Figure 1 shows an example of a reliability diagram from synthetically generated scores for a binary classification problem. Along the x-axis we have different groups of examples binned by model score. For example, the left-most bin contains examples with score less than 10% for being positive and the rightmost corresponds to examples with scores in the 90-100% range. The y-axis corresponds to the proportion of examples that were gold-labeled positive. The diagonal line marks the height of a blue bar for perfectly well-calibrated scores. The height of the blue bars correspond to the actual accuracy of scores within that bin and the red portion corresponds to the gap in accuracy between the actual scores and perfectly well-calibrated ones. From this plot we can see that our classifier is well-calibrated for scores that are less 50%, but poorly calibrated for those above 50% (high scores should be strongly discounted)[2].

The choice of number of bins, M, and where the bin boundaries lie are parameters chosen in advance. Increasing the number of bins means we get a higher resolution picture of how well-calibrated the classifier is, at the price of having noisier estimates of the true height of each bin. In our experiments, we use reliability diagrams with ten equally spaced bins.

Expected Calibration Error. Although reliability diagrams are excellent tools to understand how well-calibrated a model is across a range of predicted probabilities, they do not offer a single, hard metric to decide between two different calibration models. *Expected Calibration Error* (ECE) is one such metric that summarizes a reliability diagram into a single number. Equation (2) defines ECE as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{2}$$

B_m is the set of all $\langle example, class \rangle$ pairs that fall in the m_{th} bin. Following the notation in [2], $\text{acc}(B_m)$ is the proportion of $\langle example, class \rangle$ pairs in B_m that were of the correct class – the height of the bar in the reliability diagram. $\text{conf}(B_m)$ is the average $\hat{p}$ for examples in bin B_m (the mean of bin boundaries if scores are uniformly distributed within the bin). ECE is just the average of the *gap* between the actual proportion of positives and that achieved by a well-calibrated classifier, weighted by the number of elements in each bin. Figure 1 depicts these gaps by red bars.

$$\text{Unweighted ECE} = \sum_{m=1}^{M} \frac{1}{M} |\text{acc}(B_m) - \text{conf}(B_m)| \quad \text{s.t.} |B_m| > 0 \tag{3}$$

Unweighted ECE is another useful metric (3). This is just macro-averaged calibration error, ignoring the number of elements in each bin. This can be very useful in cases where there are few high-scoring examples but it is imperative that the model be confident in its predictions for those few examples. Imagine a system that diagnoses patients as having breast cancer. Most patients are negative and are predicted to have low probability by the model. However, when a patient is given a 90% likelihood of having cancer, we want to be very confident that they indeed are 90% likely to have breast cancer, to avoid causing them unnecessary stress. If we selected a calibration method based on (weighted) ECE, the metric would be dominated by the lower bins, which contained far more examples. Selecting a calibration method based on unweighted ECE would be more appropriate for this scenario, since gaps in the higher bins would count equally as those in the lower bins, disregarding the number of elements in each.

$$\text{MCE} = \max_{m \in \{1,\ldots,M\}} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{4}$$

Maximum Calibration Error (MCE) is yet another metric derived from the reliability diagram (Equation (4)) and is meant to capture the worst-case calibration error rather than the average. One drawback of MCE is that it is much less stable than ECE– low count bins can completely dominate this metric. We avoid evaluating models according to MCE for this very reason.

[2]Extending reliability diagrams to the multiclass case is straightforward. In this case, each class is treated separately when assigning to bins. If there are ten classes, then an example will be assigned to 10 (non-unique) bins based on the model probability assigned to each class for that example. This is equivalent to aggregating over 10 one-vs-all reliability diagrams, treating each class in turn as positive.

3 CALIBRATION MODELS

We assume that all calibration models modify pre-trained model logits, not class probabilities. This is important for calibration methods such as temperature and Platt scaling, where if they were to operate over probabilities there is no guarantee that the calibrated scores will be in $[0, 1]$. In each model description, z is the uncalibrated and $\widetilde{z}$ is the calibrated logit. For binary classification the probability of a positive example is given by $\hat{p} = \frac{1}{1+e^{-z}}$. For multiclass classification, $\forall i \in \{1 \dots |Y|\}$, $\hat{p}_i = \frac{e^{z_i}}{\sum_{j=1}^{|Y|} e^{z_j}}$.

Parameters for the calibration models are fit by holding out a group of labeled examples, a *calibration set*, and finding settings of these parameters that yield good fit on the calibration set, where the notion of fit depends on the particular calibration model.

Here we present three popular calibration models. The first two: *temperature* and *Platt scaling* are parametric models, whereas the third *isotonic regression* is non-parametric. To get a sense of how each model operates, we present reliability diagrams for the calibrated scores of a synthetic binary classification calibration set. To construct synthetic data, we select 1000 $\hat{p}$ uniformly in the range $(0, 1)$. The label of each point is assigned by the process in (5):

$$P(y = 1|z) = \begin{matrix} \frac{1}{1+e^{-z}} & \text{if } z \leq 0 \\ \frac{1}{1+e^{-z/5}} & \text{if } z > 0 \end{matrix} \Bigg\} \tag{5}$$

In other words, the original model is well-calibrated for low-scoring examples, but poorly calibrated for high-scoring examples (the true probability is squashed toward 50%). We produce a held-out set of 10^6 examples by the same generative process – the reliability diagram for (held-out) uncalibrated scores is depicted in Figure 1.

3.1 Temperature Scaling

Temperature scaling is a simple one-parameter calibration model [2]. Model logits are scaled by $\frac{1}{T}$, where T is a positive scalar learned by maximizing the log-likelihood of the calibration set (6). When $T > 1$, the temperature is hot and logits are squashed towards 0. In this case, the model becomes less confident in its predictions (predicts scores closer to 50%). When $T < 1$, the temperature is cool, causing the model to become more confident in its predictions (surprising, but not impossible). Solving for T without explicit non-negativity constraints should also be possible – T will only drop below zero if a classifier is less accurate than chance to begin with.

$$\widetilde{z} = \frac{z}{T} \tag{6}$$

Solving for T in the binary classification case amounts to maximizing the log-likelihood of the calibration set, shown in Equation (7):

$$\max_T \sum_{i=1}^{n} y_i \log[\frac{1}{1 + e^{-z_i/T}}] + (1 - y_i) \log[1 - \frac{1}{1 + e^{z_i/T}}] \tag{7}$$

This is equivalent to solving for the coefficient of a single-feature logistic regression model where the bias term is fixed to zero – a convex optimization problem. In addition to its simplicity, temperature scaling maintains identical model accuracy. If one's decision boundary is 0.5, then the calibrated model will generate identical predictions as an uncalibrated model. The main drawback is its inflexibility: one parameter affords the modeler little leeway to capture a complicated score distribution. Figure 2 displays the reliability diagrams for synthetic temperature-scaled scores ($T = 2.43$). Temperature calibration incorrectly reduces the confidence in low-scoring examples.

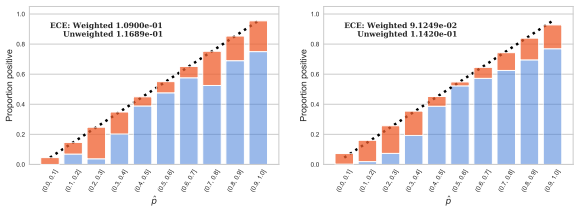

Figure 2: Reliability diagram for synthetic train (left) and held-out (right) scores after *temperature scaling*. ECE is super-imposed on each figure.

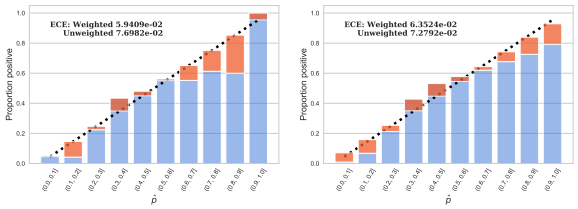

Figure 3: Reliability diagram for synthetic train (left) and held-out (right) scores after *Platt scaling*.

3.2 Platt Scaling

Platt scaling is a slight extension of temperature scaling [9]. In Platt scaling, we fit both a coefficient to scale the model logit, w, as well as a bias term, b (Equation (8)). This is equivalent to learning the parameters of a one-feature logistic regression model where our feature in the uncalibrated logit, and our target is the calibration set labels. The bias term gives the calibration model flexibility to address constant overconfidence irrespective of the predicted probability (e.g. in the case that our model always overestimates the probability of positive examples). Figure 3 displays reliability diagrams for Platt scaled scores ($w = 0.43, b = -0.50$). Platt scaling has less difficulty calibrating this score distribution than temperature calibration as the subsequent shift corrects for the "squashed" scores that are less than 0.5.

$$\widetilde{z} = wz + b \tag{8}$$

$$\begin{bmatrix} \widetilde{z}_1 \\ \widetilde{z}_2 \\ \vdots \\ \widetilde{z}_{|Y|} \end{bmatrix} = W \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{|Y|} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{|Y|} \end{bmatrix} \tag{9}$$

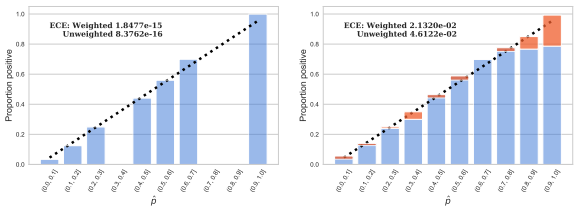

Figure 4: Reliability diagram for synthetic train (left) and held-out (right) scores after *isotonic regression* calibration.

Platt scaling can be extended to the multiclass case by *matrix scaling*, deriving calibrated logits from a linear combination of uncalibrated logits (Equation (9)). The case where the weight matrix governing these linear combinations, W, is constrained to be diagonal is known as *vector* scaling. Vector scaling amounts to Platt scaling each logit independently.

3.3 Isotonic Regression

Isotonic regression is a set of problems around learning a non-parametric function that minimizes some training objective subject to monotonicity constraints [5]. In our experiments, we only consider learning a monotonically increasing function that minimizes mean squared error between z and y on the calibration set. Equation (10) is the objective we are trying to minimize when learning isotonic regression, where y_i is the i_{th} example's class and z_i is its uncalibrated score.

$$\min_{f} \sum_{i=1}^{n} (y_i - f(z_i))^2 \quad \text{s.t.} \forall x, y \in \mathbb{R},\ x < y \rightarrow f(x) \leq f(y) \tag{10}$$

There is an efficient algorithm to solve for this function: the pooling adjacent violators algorithm (PAVA), which can recover the optimal isotone function in $O(n)$ time[3]. The optimal function turns out to be piecewise linear, where examples are implicitly binned into "plateaus." Any gaps between these bins are resolved by linearly interpolating between the plateaus.

Being a non-parametric model is both its greatest strength and weakness. This flexibility allows isotonic regression to calibrate oddly shaped score distributions, but makes it very sensitive to data sparsity, especially at the boundaries of the calibration function as test examples that fall outside of the training region will be assigned to the extreme bin. Figure 4 displays the reliability diagram for scores after fitting and applying an isotonic regression calibrator. The train ECE is extremely low, attaining almost perfect calibration. Low training ECE is a byproduct of minimizing the mean squared error objective between z and y. However, the held-out ECE it achieves is much worse, especially unweighted. The discontinuities that isotonic regression induces in the calibrated scores (Figure 5) are apparent in the bins that no longer have any counts in them.

This is not a complete set of calibration methods. Other methods include histogram binning [11] and Bayesian Binning into Quantiles [6]. However, we only consider the above Platt scaling variants and isotonic regression in our experiments as they demarcate two distinct classes of calibration methods: restrictive parametric models vs. extremely flexible, non-parametric models.

[3]By optimal, we mean that the isotonic function learned achieves minimum mean of the squared residuals over the training calibration set.

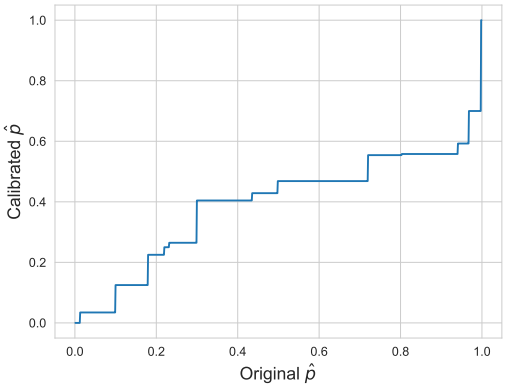

Figure 5: Calibration function learned by isotonic regression on synthetic data. The uncalibrated scores are on the x-axis and the calibrated scores they are mapped to are on the y-axis.

4 CHARCOAL GRILL SCALING

There are two potential problems with the calibration models described in Section 3. The first is that parametric calibration models, such as temperature and Platt scaling, are not afforded enough freedom to correct complicated score distributions. For the synthetic data, held-out ECE is dominated by the inability of the calibration model to adequately calibrate the score distribution. On the other hand, although isotonic regression can learn flexible calibration functions (subject to a monotonicity constraint), it is susceptible to miscalibrating extreme scores, especially when there are few training examples in this region. This is particularly disastrous when one wants to ensure that high scoring examples are well-calibrated, when poor calibration in this region leads to poor decisions affecting lives and livelihoods.

Here we propose a parametric calibration model, *Charcoal grill scaling* (CGS), that extends Platt scaling to allow for uncalibrated logits to be scaled by different degrees based on where they lie. The temperature acting on each score is derived as a weighted average of temperatures associated with different "hot spots," depending on a score's proximity to each hot spot. These hot spots can be thought of as lit coals in a barbecue grill and uncalibrated scores are particles above these coals that are pulled to a greater or lesser extent depending on which coals they are near. Although the model must be initialized with the number of such hot spots, their positions, widths, and temperatures can be efficiently optimized in the course of fitting the CGS model.

Not only does this model achieve both low weighted and unweighted ECE on the synthetic data, we believe it should be appropriate for imbalanced class and score distributions where different regions require more or less scaling. As far as we know, this is the

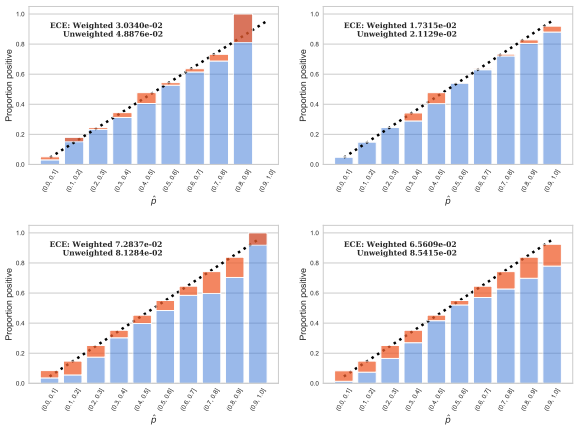

Figure 6: Reliability diagram for synthetic train (left) and held-out (right) scores after 2-cluster (top) and 3-cluster (bottom) CGS calibration.

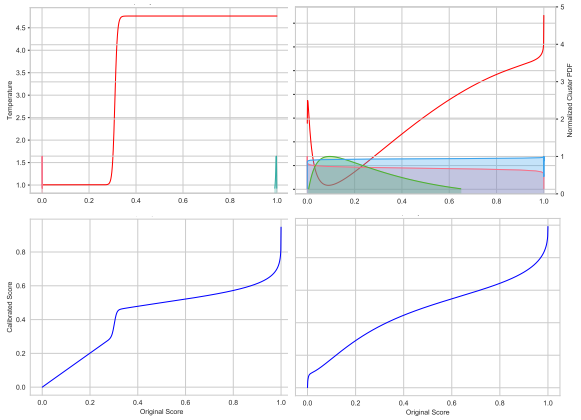

Figure 7: Calibration functions for 2-cluster (bottom left) and 3-cluster (bottom right) CGS calibration models fit to synthetic data. The top row shows the temperature acting on each score for the 2-cluster (left) and 3-cluster (right) CGS models, superimposed on representations of each cluster.

first time this model has been proposed for solving the problem of model calibration.

4.1 Model Definition

Suppose we are calibrating a binary classifier. CGS calibrates scores by Equation (11), where $T(\cdot)$ is a function that maps uncalibrated scores to the temperature they should be scaled by, and b is a scalar bias term, as in Platt scaling.

$$\widetilde{z} = \frac{z}{T(z)} + b \tag{11}$$

Suppose our CGS model has C "coals", where the c_{th} coal is parameterized by the triple: $\langle \mu_c, \sigma_c, T_c \rangle$. μ_c is a scalar for the coal location, σ_c is a scalar defining its width, and T_c is its temperature (positive scalar). $T(\cdot)$ is then defined in (12).

$$T(z) = \sum_{c=1}^{C} w_c T_c$$
$$\text{where} \tag{12}$$
$$w_c = \frac{e^{-\frac{\|z-\mu_c\|^2}{2\sigma^2}}}{\sum_{c'=1}^{C} e^{-\frac{\|z-\mu_{c'}\|^2}{2\sigma^2}}}$$

The per-coal score is determined by an (unnormalized) Gaussian distribution, and attention to each T_c (w_c) is found by passing these unnormalized scores through a softmax layer where each unit is activated by proximity to that coal. The attention weights are akin to soft cluster memberships. This can trivially be extended to the multiclass case by letting μ_c be a vector of length $|Y|$, the total number of classes[4].

[4]For simplicity, we assume that each σ_c is still a scalar, corresponding to the case of spherical Gaussians. Because this is a calibration model, we did not want to overparameterize by allowing arbitrary covariance matrices for each Gaussian. In fact, in our experiments, we fix σ_c to 1 for all clusters to simplify the learning problem.

Figure 6 shows reliability diagrams for a two and a three cluster CGS model fit to the synthetic data. Held-out ECE is slightly better than isotonic regression for the 2-cluster model (which matches the actual corruption of the data more closely). The 3-cluster model does not achieve as strong calibration, likely because the number of clusters chosen does not match how the scores were originally perturbed. Nevertheless, the calibration functions learned by both CGS models are much smoother (Figure 7), unlike isotonic regression, which learns close to a step function.

4.2 Optimization

Clusters are initialized with σ and T close to 1 (perturbed slightly), and μ distributed uniformly in $[-6, 6]$. We then fit these parameters by iteratively updating each set of parameters in turn by a single, bounded, quasi-newton step until loss has not decreased by more than 10^{-8}. We used the scipy bounded LBFGS routine for this[5]. One benefit of this method is that positivity bounds on T and σ can be explicitly passed to the method, ensuring that temperature and cluster width will both remain positive. We achieved much better success fitting the parameters with this approach than by batch or stochastic gradient descent[6]. In practice, fitting a CGS model is fast. Each CGS model took less than a minute to fit on all datasets.

Enforcing Monotonicity. Monotonic calibration functions are desirable since the ranking of examples by their calibrated scores is identical to the ranking according to their uncalibrated scores – one can always adjust the decision threshold to achieve identical 0-1 loss to the uncalibrated scores. To encourage learning a smooth

[5]https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin_l_bfgs_b.html

[6]We were not able to achieve similar ECE as Platt scaling with a single cluster CGS model, even after sweeping over learning rate for most of the gradient descent optimizers offered in pytorch. Another drawback is that we would need to somehow explicitly enforce the positivity constraints, e.g., by projecting T and σ to the positive reals. The bounded LBFGS routine in scipy allows one to explicitly enforce positivity constraints.

calibration function, we include a regularization term in our objective, defined in (13), to avoid learning a non-monotonic calibration function when calibrating binary classifiers. R defines a penalty on the scale of T and σ, and its weight in the total loss is governed by a scalar, λ.

$$
\begin{aligned}
R_{T-} &= \sum_{c=1}^{C} \left[\frac{1}{T_c} - 1 \right]_+^2 \\
R_{T+} &= \sum_{c=1}^{C} [T_c - 1]_+^2 \\
R_{\sigma} &= \sum_{c=1}^{C} (\frac{1}{\sigma_c})^2 \\
R &= R_{T-} + R_{T+} + R_{\sigma}
\end{aligned}
\tag{13}
$$

If temperatures are too extreme between clusters, or cluster widths are too small, then the calibration function may not necessarily be monotonic. Encouraging T to be close to 1, and σ to be large is one way to avoid learning non-monotonic calibration functions.

We begin optimization by setting the weight on this regularization term to be small, $\lambda = 10^{-2}$. If model training has converged, but the function is not monotonic[7], then we scale this term by 10× and continue optimization. We increase λ until training has converged and the calibration function is (empirically) monotonic. For calibrating multiclass models we do not enforce monotonicity, as there is no general definition of a monotone function for domains $\mathbb{R}^{n>1}$. In this case we set $\lambda = 10^{-2}$ to ensure a minimum smoothness to the calibration function, but this parameter can be tuned specifically for each downstream task.

5 DATASETS

We evaluate CGS on the synthetic dataset described in Section 2 as well as standard multiclass image recognition datasets, and two binary classification tasks in the financial domain. We use pretrained models whenever possible. Although we designed CGS to better calibrate imbalanced class and score distributions, we also evaluate on standard datasets with balanced classes for comparison to existing models. The multiclass classification setting may also be more difficult for CGS to calibrate, since the logit vectors lie in a space whose dimensionality is the same as the number of classes. Learning the proper place to situate the cluster centroids could be difficult.

5.1 Binary Tasks

5.1.1 Stocknet. Stocknet is a model to predict long-term price movements for an equity given tweets about the stock [10]. Classes were (artificially) balanced by treating examples with price moves $\leq -0.5\%$ as negative and moves $> 0.55\%$ to be positive examples. Price moves between these two thresholds were discarded.

We retrained Stocknet using the released training script, and found it to be relatively well-calibrated. We use the development set to fit calibration models (656 examples) and the test set to evaluate held-out calibration (1,008 examples).

5.1.2 Credit Card Fraud Detection. We also compare CGS at calibrating a model trained to predict fraudulent credit card transactions [1]. Predictions are made based on 28 anonymous features representing the user associated with the transaction, along with the amount of the transaction. We fit a 200-tree random forest classifier on 80% of examples[8]. The remaining 20% of examples were divided evenly between a calibration and evaluation set.

This dataset exhibits an enormous class imbalance, where only 0.172% of examples are true instances of credit card fraud out of a total of 284,807 instances. The random forest classifier achieved a 5-fold cross validation F1 score of 86%.

5.2 Multiclass Tasks

Although CGS was designed to calibrate models with unevenly calibrated scores (possibly caused by high class imbalance), we also evaluate on standard image recognition datasets, all framed as multiclass classification tasks.

5.2.1 Image Recognition Datasets. We calibrate on the validation sets of bitmap digit recognition in MNIST [4], naturalistic digit recognition in SVHN [7], and the CIFAR-100 image recognition dataset [3]. There are 10,000 examples in MNIST, 26,032 examples in SVHN, and 10,000 examples in the CIFAR-100 validation sets. CIFAR-100 is potentially the most challenging task for CGS to calibrate as there are 100 classes, meaning that CGS must learn a 100-dimensional mean for each cluster. Nevertheless, Guo et. al. [2] found that temperature scaling, a calibration model with a single parameter, achieved consistently low ECE when calibrating a wide variety of neural classifier architectures trained on the CIFAR-100 dataset.

Examples are shuffled and split evenly between train and held-out sets for each of these datasets. We used the pretrained models provided by https://github.com/aaron-xichen/pytorch-playground to score each example.

6 RESULTS

We evaluated the baseline calibration models presented above along with CGS where number of clusters was varied from 1 to 5. We also evaluated a variant of CGS that removes the bias term b. A 1-cluster + no-bias CGS model is equivalent to temperature scaling, whereas 1-cluster + bias CGS model is equivalent to Platt scaling.

6.1 Synthetic

We first evaluated all models on the synthetic data with corrupting temperature $T = 5$ (Table 1).

CGS models with more than one cluster outperform Platt scaling on both weighted and unweighted ECE. However, these more expressive CGS models achieve slightly worse weighted ECE than isotonic regression, but outperform isotonic regression when the number of clusters is large. This is a little strange as a CGS model

[7] We empirically check this by 10,000 uncalibrated logits uniformly distributed in $[-6, 6]$.

[8] We selected the number of trees to maximize macro-averaged cross-validation F1, and trained a model using the scikit-learn implementation: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Table 1: Weighted and unweighted 10-bin ECE (%) on synthetic data where high-scoring logits are compressed by $T = 5$ before sampling labels.

	ECE[Wted]	ECE[Unwted]
original	9.09	9.09
temperature	9.03	11.17
platt	5.85	7.34
isotonic	**1.52**	4.39
CGS-1-nobias	9.03	11.12
CGS-2-nobias	9.03	11.12
CGS-3-nobias	2.26	5.00
CGS-4-nobias	9.03	11.12
CGS-5-nobias	1.87	3.82
CGS-1-withbias	5.85	7.35
CGS-2-withbias	2.81	4.28
CGS-3-withbias	3.17	4.54
CGS-4-withbias	2.98	4.96
CGS-5-withbias	2.22	**2.69**

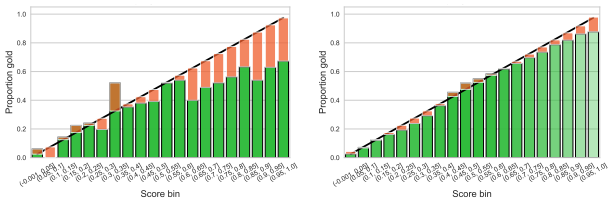

Figure 8: Reliability diagrams of uncalibrated held-out scores (left) and scores after 5-cluster CGS calibration (right).

with a few clusters should be able to properly calibrate the synthetic score distribution (e.g. place a higher temperature cluster with positive mean and a cluster with $T = 1$ with negative mean). Figure 8 displays the 20-bin reliability diagram before and after calibration with the 5-cluster CGS model (with bias).

Varying the Degree of Corruption. We also varied the temperature of corrupted logits for high-scoring examples in $\{1.1, 1.5, 2, 3, 5, 10\}$, and compared calibration models by ECE as a function of how poorly calibrated the original scores were. Held-out 10-bin weighted and unweighted ECE is shown in Figure 9. More expressive models are expected to achieve worse calibration than simpler models in the regimes where model scores are well calibrated to begin with. We find that isotonic regression achieves consistently lower weighted ECE than the other calibration methods. However, for high corrupting temperatures 5-cluster CGS models achieves the lowest unweighted ECE.

6.2 Binary Tasks

Table 2 displays both weighted and unweighted ECE after binary classifier calibration for the Stocknet and credit card *fraud* classifiers. The unweighted ECE achieved by CGS is on par with temperature scaling when fitting only a single cluster – this is unsurprising. There are no gains by fitting more clusters. Weighted ECE, on the other hand, can benefit from more clusters, although the reduction in calibration error is slight for Stocknet.

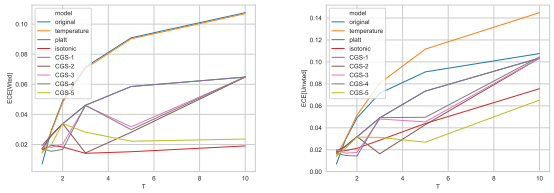

Figure 9: Held-out weighted (left) and unweighted (right) ECE for each model as a function of temperature corrupting scores. All CGS models include a bias term.

Table 2: Held-out 10-bin ECE (%) on binary classification datasets.

	ECE[Wted]		ECE[Unwted]	
Task	stocknet	fraud	stocknet	fraud
original	8.87	0.032	24.53	19.16
temperature	6.23	0.13	**6.94**	**16.81**
platt	9.60	0.038	12.88	29.32
isotonic	11.14	0.031	28.90	37.41
CGS-1-nobias	6.22	0.031	**6.94**	39.72
CGS-2-nobias	**6.16**	0.031	17.82	30.08
CGS-3-nobias	6.75	0.17	31.88	30.89
CGS-4-nobias	6.88	0.032	26.03	27.85
CGS-5-nobias	6.90	0.031	24.69	36.52
CGS-1-withbias	9.62	0.13	24.01	**16.80**
CGS-2-withbias	9.48	0.11	25.49	24.81
CGS-3-withbias	9.51	**0.026**	25.50	24.81
CGS-4-withbias	10.42	0.17	27.40	24.36
CGS-5-withbias	10.02	0.074	25.28	20.91

6.3 Multiclass Tasks

Table 3 displays the 10-bin weighted and unweighted ECE of models at calibrating models trained for each image recognition task. Isotonic regression tends to perform poorly on these tasks, but CGS models occasionally outperform the other parametric calibration methods. The relatively poor performance of isotonic regression and matrix scaling is also observed by [2].

7 DISCUSSION

Calibrating Synthetic Data. The synthetic scores dataset was designed to reflect score distributions produced by models trained with unbalanced class distributions – model is well calibrated for low-scoring examples, but less so for high-scoring ones. It was also designed such that a CGS model with few clusters should effectively calibrate this distribution. Although CGS models outperform Platt scaling, they perform similarly to isotonic regression. Assuming a sufficiently large calibration set, isotonic regression may be appropriate for properly calibrating even oddly-shaped score distributions. Nevertheless, CGS has the benefit of learning a smooth calibration function, unlike standard isotonic regression.

Table 3: Held-out 10-bin ECE (%) on multiclass image recognition datasets.

	ECE[Wted]			ECE[Unwted]		
Task	MNIST	CIFAR	SVHN	MNIST	CIFAR	SVHN
original	0.11	0.37	0.26	2.94	16.97	**1.89**
temperature	0.053	0.024	0.16	**2.42**	0.86	3.21
vector	0.053	0.019	0.15	4.36	0.79	2.92
matrix	0.13	0.66	0.14	6.41	35.68	2.84
isotonic	1.11	0.27	1.06	14.45	11.00	10.74
CGS-1-nobias	0.053	0.025	0.15	**2.42**	0.92	3.17
CGS-2-nobias	0.084	0.025	0.13	6.22	0.92	2.57
CGS-3-nobias	0.067	0.024	0.14	5.11	0.86	2.87
CGS-4-nobias	0.061	0.022	0.16	4.08	0.80	3.08
CGS-5-nobias	0.061	0.024	0.16	3.43	0.86	3.13
CGS-1-withbias	0.056	0.017	0.15	3.14	0.92	2.80
CGS-2-withbias	0.066	0.018	0.14	4.38	1.03	2.49
CGS-3-withbias	0.068	0.014	**0.13**	4.71	0.95	2.41
CGS-4-withbias	0.066	**0.013**	0.15	5.21	**0.53**	2.75
CGS-5-withbias	**0.039**	0.018	**0.13**	2.89	1.30	2.81

Benefit of CGS in Multiclass Setting. We also see benefit for fitting more expressive parametric calibration models in improving weighted ECE on image classification datasets, which were constructed with balanced classes. Isotonic regression performs poorly in this domain because each class logit is calibrated independently, potentially shifting the predicted class. Matrix scaling also tends to perform badly because of the larged number of parameters overfitting to the calibration set ($|Y| \times (|Y| + 1)$). Guo et. al. [2] showed that CIFAR-100 and SVHN network calibration can be effectively performed by temperature scaling. Our experiments suggest that calibration of multiclass classifiers may also benefit from varying the temperature applied to each region in logit space.

Where is CGS Appropriate? For binary classification, CGS is most effective at calibrating the credit card fraud scores. It is difficult to compare models based on unweighted ECE for this dataset, as there are very few examples in the higher score bins – only 53 out of over 28,841 examples in the evaluation set achieved score higher than 0.1. It is also striking that temperature scaling actually worsens model calibration, likely because over 99% of examples were assigned scores less than 0.1. One persistent difficulty is learning the parameters of CGS. This is evident in the high variance in performance across different number of clusters.

On the other hand, temperature scaling performs best on small and balanced calibration sets like Stocknet. On this dataset, CGS provides little reduction of ECE, and considering more expressive calibration models tends to overfit (e.g., the bias term in Platt scaling or isotonic regression).

8 CONCLUSION

We introduced a new variant of Platt scaling, CGS, designed to smoothly calibrate models with unevenly calibrated score distributions. These distributions can arise from models trained on very imbalanced classes, such as anomaly detectors. We compared the performance of CGS to existing calibration methods, and show that CGS is surprisingly effective at calibrating multiclass classifiers. In subsequent work, we plan to make the CGS optimization procedure more robust to cluster initialization.

REFERENCES

[1] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. 2015. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 159–166.

[2] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th ICM*, Vol. 70. JMLR, 1321–1330.

[3] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. University of Toronto.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[5] Patrick Mair, Kurt Hornik, and Jan de Leeuw. 2009. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of statistical software* 32, 5 (2009), 1–24.

[6] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[7] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[8] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 625–632.

[9] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10, 3 (1999), 61–74.

[10] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *ACL*, Vol. 1. 1970–1979.

[11] Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 694–699.