Unsupervised Recalibration

Albert Ziegler  
Semmle Inc., Blue Boar Court  
9 Alfred Street  
Oxford OX1 4EH, United Kingdom

Paweł Czyż  
St Hugh’s College, University of Oxford,  
St Margaret’s Road,  
Oxford OX2 6LE, United Kingdom

ALBERT@SEMMLE.COM  
PAWEL.CZYZ@ST-HUGHS.OX.AC.UK

Editor: Francis Bach, David Blei, and Bernhard Schölkopf

Abstract
Unsupervised recalibration (URC) is a general way to improve the accuracy of an already trained probabilistic classification or regression model upon encountering new data while deployed in the field. URC does not require any ground truth associated with the new field data. URC merely observes the model’s predictions and recognizes when the training set is not representative of field data, and then corrects to remove any introduced bias.

URC be particularly useful when applied separately to different subpopulations observed in the field that were not considered as features when training the machine learning model. This makes it possible to exploit subpopulation information without retraining the model or even having ground truth for some or all subpopulations available.

Additionally, if these subpopulations are the object of study, URC serves to determine the correct ground truth distributions for them, where naive aggregation methods, like averaging the model’s predictions, systematically underestimate their differences.

Keywords: Calibration, Post-processing, Unsupervised Learning, Maximum Likelihood Estimation, Brier score

1. Introduction

The life cycle of a classification or regression model normally consists of two distinct phases:

1. In the development phase, we select the model’s architecture, tweak its parameters and possibly evaluate it according to some input data. Ground truth for this training data is known. This task is usually performed by a ML engineer in their lab. When they evaluate the (partially or fully constructed) model, they do so in order to learn more about the model.

2. In the application phase, the model is fixed and gets applied to some field data. Ground truth for that field data is unknown. This task might still be performed by an ML engineer, or alternatively by a field operative with no background in how the

---

*. Research supported by Semmle Inc.
1. This includes training data as well as any validation, test or holdout data.
model works. When they evaluate the model, they do so in order to learn more about the world.

These phases may be alternated iteratively, sometimes in rapid succession like in online learning, in order to update the model as more input data becomes available. In some applications, field data of the previous iteration will in time get augmented by ground truth to become the input data of the subsequent iteration. In other applications, the application phase may be completely decoupled from the construction phase, for example if model deployment consists in the publication of a classification technique in a textbook.

1.1 Learning during application

Most of machine learning literature focuses on the development phase, helping ML engineers to fashion or update models that fit the ground truth in a generalizable way. But in practice, many models spend most of their lifetime in the application phase, and just observing the model’s predictions in this phase is informative and can lead us to improve the model without any need to access ground truth.

In particular, very often we are able to observe the model’s predictions in the application phase differing slightly according to some categorical property of the samples that was not used as a feature for the model. This may have been due to any of the following reasons, and often a combination of them:

- Some of the categories in the field data were not represented in the input data.
- The relevance of the property in question was not anticipated during development.
- There was insufficient input data for (some of) the different categories, and some may have been missing from the input data altogether.
- The property was unavailable at development time, or measuring would have been too resource intensive.
- The model was developed to function in a more general setting where the categories may not be available.

A typical example would be a fast-but-imprecise medical test. During application, practitioners might identify possible risk factors for which the model’s predictions are, on average, higher. A gold standard test could help quantify the influence of those risk factors, but is often too resource expensive to use on a large scale.

There are two possible reasons for this observed difference in predictions: either the different categories have different ground truth distributions, or the relationship between ground truth and model predictions is fundamentally different for the different categories. We contend that in many cases, it is reasonable to assume that the former is the driving factor behind the observed differences between categories.

However, a straightforward extension to the model to pull in the property in question is not usually possible, for any of the following reasons, and often a combination of them:

- Ground truth for field data may not be known.
Unsupervised Recalibration

- The classifier may be a black box (e.g. because it was produced by a third party).
- The classifier is of such a form that it may not easily be extended with new features.
- Re-training the classifier is too difficult, either conceptionally or computationally.
- The amount of data is insufficient for some or all categories.

Nevertheless, we will often observe classifier predictions for samples from the same category to be more homogeneous than all predictions as a whole. This indicates a correlation between ground truth and category. Because the model has untapped potential regarding this categorical property, it will tend to underestimate the actual differences (Theorem 3). By estimating and accounting for the real correlation, classifier predictions on individual samples as well as on the group as a whole can be improved.

Since the correlation is underestimated without specific post-processing, a particular important application of URC is when the differences between the categories are the main focus of the model’s current application.

In the medical example above, the model may be used to quantify the possible risk factors. Using the naive estimation of risk factors by taking the average model prediction for each category as an estimator for the incidence rate would underestimate the influence of the category.

1.2 Relationship to existing techniques

Calibrating a classifier in order to make its predictions match the actual distribution of ground truth is an established post-processing technique (Gebel, 2009). It consists of tweaking the classifier predictions in a generalizable way to match the observed ground truth distribution in every case (where a “case” is usually a group of samples with similar predictions). Our technique can be seen as a kind of calibration, but one that occurs during the application phase, where the ground truth distribution can not be directly observed.

The proposed method of unsupervised recalibration also bears some resemblance to online learning (Rosasco and Poggio, 2015). In particular, techniques have been proposed to allow progressive learning classifiers to learn new classes on the fly (Venkatesan and Er, 2016), and indeed one of the use cases for URC is to make predictions on classes not (or insufficiently) encountered in training. As before, the main difference is that classical online learning relies on the extension of input data by learning the ground truth for more samples. Occasionally such data becomes available shortly after an prediction is made, however, often it is either impossible to come by or would require significant resource investment. So we do not rely on it.

Finally, we infer the ground truth distribution for each category from the different model predictions for samples from this category. Each sample yields one indication for a possible ground truth distribution. From these many estimates we do obtain a better estimate for the ground truth distribution for the category than could be obtained by any single estimate alone, yet the nature of the inter-correlation between the single estimates precludes using ensemble learning (Rokach, 2010) or distribution conflation (Hill, 2011) techniques.
2. Outline

In section 3, we will introduce the heart of URC: a technique to recalibrate a classifier that was calibrated on a biased input set without direct information about that bias (“global unsupervised recalibration”). In section 4 we will make use of this technique to improve general classifiers based on context (“local unsupervised recalibration”). Together, these sections form algorithm 1.

In section 5, we will discuss how to extend this approach to regression problems. We will see how the techniques described in this paper improve a classifier’s performance in practice in section 6 and close with a discussion of when not to use URC in section 7.

3. Global unsupervised recalibration

Assume you have a categorical random variable $X \in \{1, ..., n\}$, which describes the labels, and a random variable $C \in \mathbb{R}^n$, which describes a probabilistic classifier predicting $X$. Each dimension $C_i$ refers to the predicted probability that $X$ be $i$. We assume no knowledge about the inner workings of $C$, but know it was devised based on some training population $T$, which we’ll identify with its characteristic function, a random variable $T \in \{0, 1\}$ (where 0 means “member of the training population” and 1 means not).

If the classifier was successfully trained and calibrated on the training set $T$, we expect that for every realization of $C$, $C_i$ describes the probability that $X = i$ given the value of $C$ and $T = 1$. I.e.

$$C_i|_{T=1} = P(X = i \mid C, T)|_{T=1}. \tag{1}$$

If the training population $T$ is unbiased for $X$, i.e.

$$P(X = i \mid T) = P(X = i), \tag{2}$$

and $C$ has not picked up any features specific to $T$, then equation 1 implies $C$ is correctly calibrated overall, i.e.

$$C_i = P(X = i \mid C). \tag{3}$$

But in many cases, the input set will be biased in some way. Often, this is even deliberate in order to facilitate training, for example through stratification (Särndal et al., 2003).

Machine learning engineers usually make the implicit assumption that this bias is “tame”. The sample may be biased with respect to $X$, but the sample does not differ from the overall population with respect how $X$ and $C$ interact: conditioned on each class $X = i$, the distribution of $C$ is assumed to be identical in the training set and the real world. In other words, $C$ and $T$ are relatively independent given $X$.

In many applications, this assumption is highly plausible since it already holds for the features used to compute $C$: for each class $X = i$, the distribution of the features for members of this class does not differ between training set and overall population.

We will make this assumption too: in the following, let for any event $\mathcal{C}$ in the $\sigma$-algebra generated by $C$

$$P(\mathcal{C} \land T \mid X = i) = P(\mathcal{C} \mid X = i) \cdot P(T \mid X = i). \tag{4}$$
Unsupervised Recalibration

We suggest to see the biased training set problem as a miscalibration problem: If \( T \) is biased for \( X \) (equation 2 does not hold) and \( C \) is calibrated for \( X \) on \( T \) (equation 1 holds), \( C \) cannot be calibrated for \( X \) overall (equation 3 does not hold).

Moreover, this miscalibration is not of a form that most established parametric calibration techniques assume. This is because it stems not from the classifier being either simply overconfident (Platt et al., 1999) or underconfident (Kull et al., 2017), but from the classifier systematically underestimating the probability of \( X = i \) in case \( P(X = i) > P(X = i|T) \) or systematically overestimating it in case \( P(X = i) < P(X = i|T) \).

Biased training sets are a known problem, and previous approaches aimed at quantifying and minimising that problem. For example, Shahrokh Esfahani and Dougherty (2013) suggest a sample stratification strategy that yields the minimax error considering an unknown bias. However, this error is still considerable, which is why we will aim at correcting it entirely.

3.1 Global unsupervised recalibration for known bias

If the bias is known, recalibration becomes applying a simple formula.

**Lemma 1** Define unnormalized probabilities as

\[
\bar{P}_i := C_i \cdot \frac{P(X = i)}{P(X = i | T)} \tag{5}
\]

Then

\[
P(X = i | C) = \frac{\bar{P}_i}{\sum_j \bar{P}_j} \tag{6}
\]

**Proof** Direct calculation.

3.2 Global unsupervised recalibration for unknown bias

Due to lemma 1, all we need is a sensible estimate for the true distribution of \( X \) in the population. A first pass might use the naive estimator:

**Definition 2** An unbiased estimator for the expected value \( E(C_i) \) is called a naive estimator for \( P(X = i) \).

However, while useful for predicting the direction of the bias given enough data, this will normally underestimate the magnitude of the bias and will not approach the true value as the sample size increases.

**Theorem 3** If

\[
P(X = i) < P(x = i | T), \tag{7}
\]

then

\[
P(X = i) < E(C_i) < P(X = i | T). \tag{8}
\]

The corresponding statement with \( > \) also holds.
The inequality holds pointwise for each $C_i$, as $C_i$ and $T$ are relatively independent given $X$ and $P(X = i) < P(X = i | T)$ (equation 4). This proves the first part of equation 8. For the second, we continue the chain of inequalities through

$$
\mathbb{E}(C_i) = \int P(X = i | C_i \wedge T) \, dC_i \tag{13}
$$

$$
\mathbb{E}(C_i) = \int P(X = i | C_i \wedge T) \, dC_i \tag{14}
$$

$$
\mathbb{E}(C_i) = \int P(X = i | C_i \wedge T) \, dC_i \cdot \frac{\int P(T \wedge C_i) \, dC_i}{P(T)} \tag{15}
$$

$$
\mathbb{E}(C_i) = \int P(X = i | C_i \wedge T) \cdot \frac{P(T \wedge C_i)}{P(T)} \, dC_i \tag{16}
$$

$$
\mathbb{E}(C_i) = \int P(X = i | C_i \wedge T) \cdot d(C_i | T) \tag{17}
$$

$$
\mathbb{E}(C_i) = P(X = i | T) \tag{18}
$$

Here, the inequality holds because $P(C_i \wedge T)$ is strictly increasing in $C_i$, as is $P(X = i | C_i \wedge T)$.

If the classifier is very accurate, i.e. if $C_i(1 - C_i) \approx 0$, then the naive estimator will be reasonably close to the desired $P(X = i)$ (and the classifier will already be reasonably well calibrated to the field data). Otherwise, an alternative is needed.

Our alternative centers around partitioning classifier predictions into a finite number of clusters and analysing how often each cluster appears. We will then work backwards to determine which ground truth distribution might have caused these observations. The partition is a hyperparameter of URC.

**Definition 4** Call a family of sets $(A_i)_{i=1...n} \subseteq \mathbb{R}^n$ a partition for $C$ if $P(C \in A_i \cap A_j) = 0$ for $i \neq j$ and $P(C \in \bigcup_{i=1}^n A_i) = 1$.

For a given partition $A$ of $C$, define

$$M_A = (P(C \in A_j \mid X = i \wedge T))_{i,j=1...n},$$

where each element $m_{i,j}$ of $M_A$ is the probability the classifier predicts an element of partition $j$ given an example of category $i$ in the training set. $M_A$ therefore encodes the distribution of predictions conditional on the ground truth. Define further

$$\vec{v}_A = (P(C \in A_j))_{j=1...n},$$
where each element $v_{A_j}$ of $v_A$ is the probability the classifier predicts an element of partition $j$ in the field.

The matrix $M_A$ is computed during the development phase. The vector $v_A$ is observed in the field. These two include all the information URC requires to estimate the ground truth distribution in the field data (although a few extra summaries on the training data might be useful for regularisation, see remark 9, part 1).

The URC equations work for any partition\(^2\) of $C$. For reasons of numerical stability, it is sensible to choose a system of sets with similar probability and low variation. In the binary classification case, we suggest taking intervals which appear as equally likely from the training set. For a partition into $m$ intervals, this would mean for all $i \leq m$:

$$A_i = \left\{ (x_1, x_2) \mid x_1 + x_2 = 1 \land \frac{i-1}{m} < P(C \leq x_1 \mid T) \leq \frac{i}{m} \right\}$$  \hspace{1cm} (21)

**Lemma 5** Let $(A_i)_{i=1...n} \subseteq \mathbb{R}^n$ be a partition for $C$ and let

$$\vec{p}_x = (P(X = 1), \ldots, P(X = n)).$$

Then:

$$M_A \cdot \vec{p}_x = \vec{v}_A$$  \hspace{1cm} (22)

**Proof** Because of relative independence (equation 4),

$$P(C \in A_j \mid X = i \land T) = P(C \in A_j \mid X = i).$$

So the equation follows from case distinction on $X = x_1 \lor \ldots \lor X = x_n$.  \hspace{1cm} \(\blacksquare\)

Lemma 5 is an $n \times n$ system of linear equations relating the ground truth (which is not directly observable) to the model predictions (which are directly observable). $M_A$ is usually full rank (depending on the choice of partition), and then we could solve for $\vec{p}_x$.

But this is of limited use in practice: If the classifier is not very accurate, the condition number $\kappa$ of the above equation will be very high. Since sample limitations will usually leave at least some uncertainty when estimating the probability of $P(C \in A_j)$, lemma 5 will not often narrow down $P(X = i)$ in a meaningful way. In the case we extend our approach to partitions with more than $n$ members, this uncertainty will even lead to the (now over-determined) system normally being unsolvable when using the approximate values for $\vec{v}_A$.

However, it does give rise to an optimization problem. Instead of solving for $P(X = i)$ directly, we can judge how likely a series of observations was for a candidate $P(X = i)$. Define a loss function as follows.

**Definition 6** For an unbiased sample $S$ and a partition $A$ for $C$, define a function $L_{\loglikelihood} : [0, 1]^n \to \mathbb{R} \cup \{\infty\}$ by

$$L_{\loglikelihood}(\vec{p}) = -\log(B(|S|, (|S \land C \in A_i|)_{i=1...n}, M_A \cdot \vec{p})),$$  \hspace{1cm} (23)

where $B(m, \vec{k}, \vec{p})$ is the multinomial mass function, i.e.

$$B(m, \vec{k}, \vec{p}) = \binom{n}{k_1, \ldots, k_n} p_1^{k_1} \cdot \ldots \cdot p_n^{k_n}$$  \hspace{1cm} (24)

\(^2\) In fact, they generalize to a partition into more than $n$ sets.
Proposition 7 If $M_A$ is full rank, $L_{\text{loglikelihood}}$ has a single global minimum, which for $|S| \to \infty$ converges against $\vec{p}_x$.

Proof The minimum of the multinomial function $B(m, \vec{k}, \vec{p})$ is attained at $\vec{p} = \vec{k}/m$, so in the limit case, where

$$|(S \& C \in A_i)|/|S| \to P(C \in A_i),$$

we approach $\vec{p}_x$, the solution of the equation in Lemma 5.

Solving for minimal $L_{\text{loglikelihood}}$ directly is risky due to the ill conditioned nature of $M_A$. Usually, it makes sense to combine it with a regularization loss.

Theorem 8 Let $L_{\text{regularization}}[0, 1]^n \to \mathbb{R}^+ \cup \{\infty\}$ be any $C^2$ function that fulfills $L_{\text{regularization}}((P(X = i))_{i=1...n}) < \infty$ and let $M_A$ be full rank.

The the global minimum of $L_{\text{loglikelihood}} + L_{\text{regularization}}$: 
1. ... exists and is unique with arbitrarily high probability for sufficiently large $S$.
2. ... converges in probability to $\vec{p}_x$ as $|S| \to \infty$.
3. ... is the maximum a posteriori estimate (Bassett and Deride, 2018) for $\vec{p}_x$ for fixed $T$ and prior probabilities proportional to $e^{L_{\text{regularization}}}$.

Proof
1. Let $p < 1$, we need to find an $k$ such that for $|S| \geq k$, the probability for unique existence of the minimum is at least $p$. Since $L_{\text{regularization}}$ is continuous, there is a compact neighborhood $U_1$ of $(P(X = i))_{i=1...n}$ where $L_{\text{regularization}} < \infty$. By the extreme value theorem, there is a $b \in \mathbb{R}$ such that $\left(\frac{d}{dp_i}\right)^2 L_{\text{regularization}} > b$ for all $i$ and in $U_1$.

The second derivative of the multinomial density $B$ in any direction is bounded from below by $|S|$, a value which is attained for $\vec{k} = (|S|, 0, ..., 0)$ and $\vec{p} = (1, 0, ..., 0)$. So for $|S| > -b$, the sum of the two losses is convex in $U_1$.

Let $U_2 \subset U_1$ be a compact neighborhood of $\vec{p}_x$ fully contained in the interior of $U_1$ such that the diameter of $U_2$ is smaller than the smallest distance of a point of $U_2$ to the boundary of $U_1$. Let $k_0$ be such that for $|S| > k_0$, the probability of the vector of ratios $v_r = (|(S \& C \in A_i)|/|S|)_{i=1...n}$ being in $U_2$ is at least $p$.

Since $U_1$ is compact and $L_{\text{regularization}} < \infty$ on $U_1$, the regularization loss has a maximum value $mU_1$. Let $k_1 > k_0$ be such that for $|S| > k_1$, the loglikelihood loss of a vector being more than the diameter of $U_1$ away from the minimum $v_r$ is more than $mU_1$ higher than the loglikelihood loss at that minimum.

Conditioned on $v_r \in U_2$, the combined loss at $v_r$ is smaller than at any point outside of $U_1$. As the combined loss is convex on $U_1$, there exists a unique global minimum. Since the event we conditioned in has probability at least $p$, so does the existence of a unique minimum.
2. In the above, for any neighborhood \( U_3 \ni \vec{p}_3 \), let \( k_2 > k_1 \) be such that the probability of \( v_r \in U_3 \cap U_2 \) is at least \( p \). Then the global minimum is in \( U_3 \).

3. The sum of the logarithms of likelihood and prior is minimized when the product of likelihood and prior is minimized, i.e. the posterior.

We propose to approximate this minimum and take it as estimator for the desired probabilities \( \tilde{p}_3 \).

**Remark 9**

1. *In our primary use case, we investigated different candidates for \( L \) regularization.* We settled on a loss proportional to the Kullback-Leibler divergence of the candidate \( \tilde{p} \) from the distribution observed on \( T \), reasoning that the default assumption for the overall population should be one similar to \( T \).

2. While \(|A| = n\) is necessary for solving the equation in lemma 5, it is not for optimizing a function as \( 8 \). It is perfectly reasonable to use partitions \( A \) with more than \( n \) elements, and indeed, in our primary use case, we had good experiences with \( n = 2, |A| = 4 \).

4. **Local unsupervised recalibration**

Assume that when applying your trained probabilistic classifier \( C \in \mathbb{R}^n \) for a random variable \( X \in \{1, \ldots, n\} \) in the field, you encounter a discrete random variable \( P \), which you have not encountered before. It is often reasonable to assume that the different levels may have a different \( P(X = i | P = p_j) \), but that the relationship between \( X \) and \( C \) is always the same in field samples.

In that case, we can apply the global unsupervised recalibration procedure for each \( a_i \) individually, i.e. we recalibrate \( C \) on \( T \cap \{P = p_i\} \) in order to have an unbiased classifier on \( \{P = p_i\} \) (see algorithm 1).

**Algorithm 1:** Unsupervised recalibration for classification

```
partition predictions into sets using equation 21;
for each category (if separate categories are encountered in the field) do
    get posterior ground truth distribution by minimizing the loss given in definition 6 plus an optional regularization loss;
    for each sample in category do
        obtain calibrated probability by applying lemma 1;
    end
end
```

For levels of \( P \) with few encountered examples, we will stay mainly with our priors. For levels with many examples, we will converge against the true values according to theorem 8.
The recalibrated classifier is then well calibrated for each level of $P$ individually for which there is sufficient data.

5. Extension to regression

As usual, a regression model that produces a probability distribution can be calibrated by splitting the support of the distribution in $n$ intervals. The regression model is equivalent to a probabilistic classifier that gives a probability for each interval and individual regression models that give distributions within each interval conditioned on the event that the true value is in that interval. The probabilistic classifier is then recalibrated as described above.

Since such a classifier predicts an ordered categorical set, it makes sense to include a continuity component into $L_{\text{regularization}}$, i.e. $L_{\text{regularization}}$ should generally increase if $|C_i - C_{i+1}| \gg 0$.

Also, it makes sense to choose a partition based on intervals of the predicted overall value (which is a linear combination of the $C_i$). Analogous to equation 21, our suggestion would be to split into $n$ quantiles (as observed on the training data).

6. Experiments

Unsupervised calibration has the potential to improve a large range of probabilistic classifiers. To test this claim on a state-of-the-art classifier, we used Wolfram’s ImageIdentify Net V1 (Wolfram, 2017) to classify low resolution images.

The classification task was deciding whether a given image depicts a beetle or a butterfly. These categories were chosen as typical, but visually distinct orders of insects for which there is a good supply of training data available. We obtained our data by decreasing the resolution of pictures from the iNaturalist Challenge dataset (Van Horn et al., 2018). It comprises 57,742 pictures, each size reduced to six different sizes with a maximum dimension of 30, 40, 50, 75, 100 and 200 pixels respectively, while retaining the original aspect ratio.

The necessary code to obtain that dataset and replicate the following experiments is open-sourced in Ziegler and Czyż (2019).

Since the image classifier we use has been built as a general purpose image classifier not limited to beetles or butterflies, we remove all other predictions and re-normalise so that $p_{\text{beetle}} + p_{\text{butterfly}} = 1$. We are aware that this is a crude way to force a prediction, but contend that this does not detract from this classifier’s ability to serve as a proof-of-principle for the method under consideration.

The resulting classifier is not well calibrated even for a balanced training set, so we calibrate it first. This benefits the classifier before recalibration more strongly than the classifier after recalibration.

---

3. For the purpose of this article, we classified moths as butterflies as well, since together they comprise the order Lepidoptera. This avoids questions like “Are skippers moths or butterflies?” and lets us compare one biological order (Lepidoptera) against another (Coleoptera). While there are many species of moths, the majority of our pictures of lepidopterans depict actual butterflies.

4. We use Platt scaling (Platt et al., 1999), which brings down the calibration component of the Brier score on a balanced set from 3% to 0.1%.
Figure 1: *Examples where the classifier needs high resolution to correctly solve the classification task.* The spotted cucumber beetle (above) only fills a small portion of the image and is crawling over buds which at low resolutions might conceivably resemble the folded wings of a butterfly. The white-streaked prominent (below) is a moth, which generally suffer from a higher misclassification rate. Its brown-grey color is more typical for beetles than for butterflies. The example pictures in the plot have been downscaled to contain 100 pixels in their largest dimension.

At full resolution, beetles have a 86.1% chance of being identified\(^5\) correctly, and butterflies have a 87.7% chance. It proved impossible to compare these numbers with the performance of ImageIdentify on its original training set, since neither those statistics nor the training set itself are published. However, such summaries (yielding \(m_A\)) are required as parameters for unsupervised calibration. To approximate them, we use a small balanced set of 200 randomly selected beetles and butterflies images at full resolution to simulate the evaluation of the classifier on the training set. We perform unsupervised calibration on the predictions of the classifier on the downsized versions of the other images.

### 6.1 Global unsupervised recalibration

For all tested image sizes, unsupervised calibration improved the log likelihood, Brier score and accuracy of the associated hard classifier considerably. This effect was strongest at low

\(^5\) When evaluating the classifier as a hard classifier, we take as its prediction the class to which it assigns the higher probability.
resolutions where the original classifier was weakest. It was robust\(^6\) against the number of partitions and the choice of evaluation set. The results are shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>image size in pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
</tr>
<tr>
<td>negative log likelihood per sample</td>
<td>0.399</td>
</tr>
<tr>
<td></td>
<td>0.357</td>
</tr>
<tr>
<td>Brier score</td>
<td>to</td>
</tr>
<tr>
<td></td>
<td>0.126</td>
</tr>
<tr>
<td></td>
<td>0.091</td>
</tr>
<tr>
<td>calibration component of Brier score</td>
<td>to</td>
</tr>
<tr>
<td></td>
<td>0.044</td>
</tr>
<tr>
<td></td>
<td>0.008</td>
</tr>
<tr>
<td>hard classification accuracy</td>
<td>to</td>
</tr>
<tr>
<td></td>
<td>0.826</td>
</tr>
<tr>
<td></td>
<td>0.893</td>
</tr>
</tbody>
</table>

Table 1: Effects of global unsupervised recalibration. All numbers have been averaged over 100 different random choices of the evaluation set and were calculated using 4 partitions.

In all cases, unsupervised recalibration computes that the base rate for beetles is at most 17\% (it is actually 11\%), while the average unrecalibrated probability was between 23\% and 27\%. Accordingly, recalibration reclassifies some images that were previously considered beetles as butterflies. This increases the average precision for the beetles predictions considerably (32\% to 63\% for 30 pixel images, and 47\% to 76\% for 200 pixel images). Conversely, the average precision for butterfly predictions decreases only slightly (94\% to 90\% for 30 pixel images, and 98\% to 94\% for 200 pixel images). In all 100 experiments, the classification accuracy increases by at least 6\% (mean increase: 7\%) for 30 pixel images, and by at least 4\% (mean increase: 5\%) for 200 pixel images (see Figure 2).

A good way to evaluate the performance of a probabilistic classifier is the Brier score (Hernández-Orallo et al., 2011). This score can be decomposed\(^7\) into a refinement and a calibration component. Intuitively speaking, refinement measures the classifier’s ability to distinguish between samples which are highly likely to belong to one class and samples which are highly likely to belong to the other class, while the calibration component measures that these likelihoods are reported correctly.

Figure 3 shows the improvement in the Brier score. Global unsupervised recalibration does not impact a classifier’s refinement, so all improvement in the Brier score is due to the

\(^6\) In Table 1, when running the set of 100 experiments with different numbers of partitions (2, 3, 8, and 16), results never differed from the reported values by more than relative 9\% for any value and more than relative 5\% for any value other than the 30 pixels one, except for the post-calibration calibration component of the Brier score, where the low absolute values make relative differences less relevant.

\(^7\) The decomposition requires a choice for partition of the predictions. The numbers we report here have been computed using deciles.
Unsupervised Recalibration

Figure 2: *Hard classification accuracy for the original and recalibrated classifier.* The ribbons around the blue dots represent the 95% range for the values depending on different choices of evaluation set. Recalibrating the classifier on the low 30 pixel resolution still produces a more accurate result than not recalibrating the classifier on the relatively high resolution of 200 pixels.

improvement in its calibration component. This is in stark contrast to local unsupervised recalibration (see below).

6.2 Local unsupervised recalibration

Local unsupervised recalibration takes advantage of the division into subpopulations. In this case, this could entail sorting the images by identity of the photographer, or the location or season in which they were created. It is likely that restricted to each class, the unmodified classifier works similarly well on all these subpopulations, yet the different subpopulations probably have different base rates for the classes. This difference in base rates determines the strength of local unsupervised recalibration.

We want to test this relationship systematically from a neutral starting point and separate it from the global effect. So we use the full resolution images and subset the iNaturalist training data to make it balanced. Since we made sure to start with a classifier that is already well calibrated in this setting, there is nothing for global unsupervised recalibration to improve here.

We then randomly assign the remaining 12,894 images to subpopulations 1 and 2 such that the number of beetles contained in each corresponds to a set base rate. It turns out that unsupervised recalibration never hurts\(^8\), with its benefits being strongest for very unbalanced subpopulations (see Figure 4).

---

\(^8\) In other examples, however, it can hurt if the classifier started out being biased for the subpopulations, or if the amount of data is so small that the estimated base rate is unluckily far from the actual base rate.
Figure 3: Brier score before and after recalibration. The Brier score is composed of the refinement component, which is unaffected by global unsupervised recalibration, and the calibration component, which decreases considerably under recalibration. The ribbons around the blue dots represent the 95% range for the values depending on different choices of evaluation set.

In contrast to global recalibration, local recalibration improves the refinement of the classifier. Since we started with an already well calibrated classifier, the calibration component of the Brier score is always (close to) 0, while the refinement component only approaches 0 if subpopulation membership completely determines class membership.

This strongly underlines than when natural subpopulations occur which are not expected to be independent from the quantity one wishes to predict, local unsupervised recalibration is highly effective. If it is impossible or unfeasible to retrain the classifier with the subpopulation as an input feature, unsupervised recalibration has the potential to improve performance considerably.

7. Contraindications

The technique described in this article cannot be applied blindly. In particular, there are two big contraindications that should always be considered carefully. Do not apply local recalibration if any of the following holds:

1. The original classifier has a bias for the subpopulations under consideration.

   This bias would be increased with local unsupervised recalibration.
Figure 4: **Effect of local unsupervised recalibration.** Local unsupervised recalibration is most beneficial if the subpopulation differ substantially in their class membership distributions. Recalibration shown for 4 partitions, but other numbers yield highly similar results. Since the classifier was well calibrated from the beginning, the calibration component of the Brier score is close to 0, and the refinement component is close to the total Brier score.

Such a bias commonly arises from the original classifier having access to features which are a good proxy for the subpopulation. It could also arise from the training data labels being tainted (Jiang and Nachum, 2019).

2. The classification is desired to be bias free for the subpopulations under consideration. Local unsupervised recalibration will introduce such a bias.

The original classifier might not take the subpopulation into account by design. E.g. while parental income might be correlated with academic success, it is strongly contraindicated to recalibrate university admissions tests by parental income bracket, which would have the effect of preferring the more affluent applicant in the case of similar objective scores.

Additionally, it is sometimes considered desirable in a hard classifier to have similar performance on all classes. Unsupervised recalibration does not optimize for this property – in fact, it deliberately sacrifices performance on rare classes to gain improved performance on common classes. However, it is nevertheless advisable to run unsupervised recalibration in order to determine the base rate $P(X = i)$, which allows to transform the probabilistic classifier into a hard classifier in a way such that performance on all classes is maximised.
8. Summary

Unsupervised recalibration addresses two common problems in applying machine learning models:

- A model is applied in an environment where the ground truth distribution is not guaranteed to reflect the distribution in the model’s training set (i.e., the training set may exhibit an unknown bias).

- During model application, samples can be sorted into relevant subpopulations which were not taken into account to train the model (i.e., new features become available).

In these situations, unsupervised recalibration can improve classification results by a considerable margin (see sections 3 and 4). In contrast to established methods, it does not require gathering new ground truth for the new environment or subpopulations, which is often extremely costly or impossible, and without retraining the original ML model, which is sometimes costly and often impossible.

References


Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

