The Why and How of Classifier Calibration

Peter Flach, with slides contributed by *Telmo Silvo Filha* and *Hao Song*, and prepared in collaboration with *Miquel Perello Nieto*, *Meelis Kull* and *Raul Santos-Rodriguez*

`classifier-calibration.github.io/`

What I plan to cover

Setting the stage

How to measure calibration

Post-hoc calibrators in action

Some advanced topics

Concluding remarks

I will now talk about...

Setting the stage

- Taking inspiration from forecasting
- Why do we want classifiers to be calibrated?
- Common sources of miscalibration
- A first look at some calibration techniques
- Multi-class calibration

Taking inspiration from forecasting

Weather forecasters started thinking about calibration a long time ago (Brier, 1950).

- A forecast '70% chance of rain' should be followed by rain 70% of the time.

This is immediately applicable to binary classification (e.g., spam email):

- A prediction '70% chance of spam' should be spam 70% of the time.

and to multi-class classification (e.g., Fisher's iris data):

- A prediction '70% chance of setosa, 10% chance of versicolor and 20% chance of virginica' should be setosa/versicolor/virginica 70/10/20% of the time.

In general:

- **A predicted probability vector should match empirical frequencies.**

But what do we mean by 'x% of the time'?

Forecasting example

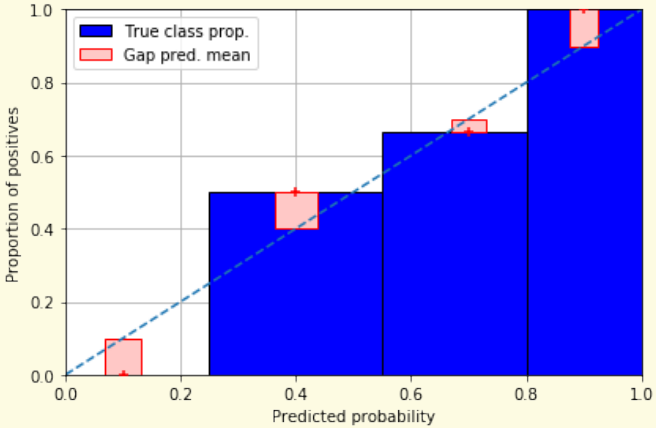
	$\hat{p}$	y
0	0.1	0
1	0.1	0
2	0.4	0
3	0.4	1
4	0.7	0
5	0.7	1
6	0.7	1
7	0.9	1

This forecaster is doing a pretty decent job:

- '10%' chance of rain' was a slight **over**-estimate ($\bar{y} = 0/2 = 0\%$);
- '40%' chance of rain' was a slight **under**-estimate ($\bar{y} = 1/2 = 50\%$);
- '70%' chance of rain' was a slight **over**-estimate ($\bar{y} = 2/3 = 67\%$);
- '90%' chance of rain' was a slight **under**-estimate ($\bar{y} = 1/1 = 100\%$).

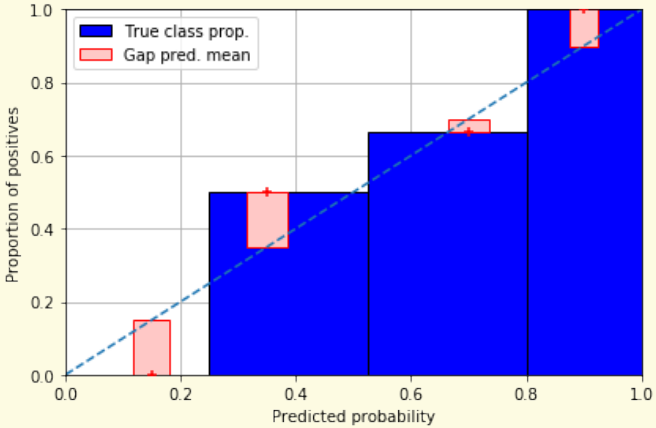
Visualising forecasts: the reliability diagram

	$\hat{p}$	y
0	0.1	0
1	0.1	0
2	0.4	0
3	0.4	1
4	0.7	0
5	0.7	1
6	0.7	1
7	0.9	1



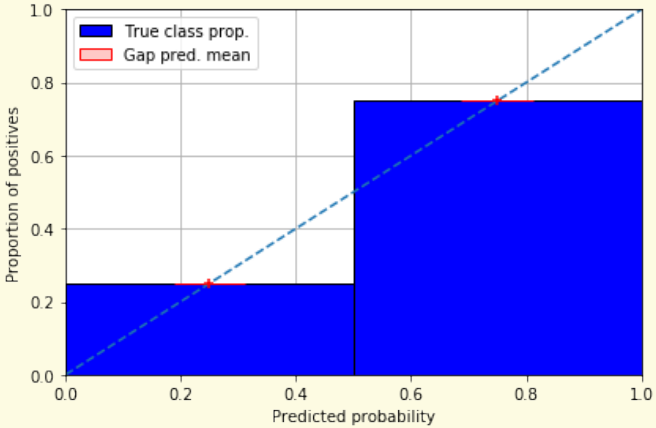
Changing the numbers slightly

	$\hat{p}$	y
0	0.1	0
1	**0.2**	0
2	**0.3**	0
3	0.4	1
4	**0.6**	0
5	0.7	1
6	**0.8**	1
7	0.9	1



Changing the bins changes the picture

	$\hat{p}$	y
0	0.1	0
1	**0.2**	0
2	**0.3**	0
3	0.4	1
4	**0.6**	0
5	0.7	1
6	**0.8**	1
7	0.9	1



Binning or pooling predictions is a fundamental notion

We need bins to **evaluate** the degree of calibration:

- In order to decide whether a weather forecaster is well-calibrated, we need to look at a good number of forecasts, say over one year.
- We also need to make sure that there are a reasonable number of forecasts for separate probability values, so we can obtain reliable empirical estimates.
  - Trade-off: large bins give better empirical estimates, small bins allows a more fine-grained assessment of calibration.

But adjusting forecasts in groups also gives rise to practical calibration **methods**:

- empirical binning
- isotonic regression (aka ROC convex hull)

Why do we want classifiers to be calibrated?

To calibrate means **to employ a known scale with known properties**.

- E.g., additive scale with a well-defined zero, so that ratios are meaningful.

For classifiers we want to use the probability scale, so that we can

- justifiably use default decision rules (e.g., maximum posterior probability);
- adjust these decision rules in a straightforward way to adapt to **changes** in class priors or misclassification costs;
- combine probability estimates in a well-founded way.

(Q: Is the probability scale additive?)
(Q: How would you combine probability estimates from several well-calibrated models?)

A bit of utility theory I

Denote the cost of predicting class j for an instance of true class i as $C(\hat{Y} = j|Y = i)$.
The **expected cost** of predicting class j for instance x is

$$C(\hat{Y} = j|X = x) = \sum_i P(Y = i|X = x)C(\hat{Y} = j|Y = i)$$

where $P(Y = i|X = x)$ is the probability of instance x having true class i (as would be given by the Bayes-optimal classifier).
The **optimal decision** is then to predict the class with lowest expected cost:

$$\hat{Y}^* = \operatorname*{argmin}_j C(\hat{Y} = j|X = x) = \arg\min_j \sum_i P(Y = i|X = x)C(\hat{Y} = j|Y = i)$$

A bit of utility theory II

In **binary classification** we have:

$$C(\hat{Y} = +|X = x) = P(+|x)C(+|+) + \big(1 - P(+|x)\big)C(+|-)$$
$$C(\hat{Y} = -|X = x) = P(+|x)C(-|+) + \big(1 - P(+|x)\big)C(-|-)$$

On the optimal decision boundary these two expected costs are equal, and hence

$$P(+|x) = \frac{C(+|-) - C(-|-)}{C(+|-) - C(-|-) + C(-|+) - C(+|+)} \triangleq c$$

This gives the **optimal threshold** on the hypothetical Bayes-optimal probabilities. It is also the best thing to do in practice – as long as the probabilities are well-calibrated!

Costs and class priors: two sides of the same coin...

Without loss of generality (for decision-making purposes) we can set the cost of true positives and true negatives to zero; $c = \frac{c_{FP}}{c_{FP}+c_{FN}}$ is then the cost of a false positive in proportion to the combined cost of one false positive and one false negative.

- E.g., if false positives are 4 times as costly as false negatives then we set the decision threshold to $4/(4+1) = 0.8$ in order to only make positive predictions if we're pretty certain.

Similar reasoning applies to changes in class priors:

- if we trained on balanced classes but want to deploy with 4 times as many positives compared to negatives, we lower the decision threshold to 0.2;
- more generally, if we trained for class ratio r and deploy for class ratio r' we set the decision threshold to $r/(r+r')$.

Cost and class prior changes can be combined into a single threshold (Flach, 2014).

...but not for multi-class classification

- For $K > 2$ classes the full cost matrix has $K(K-1)-1$ degrees of freedom.
  - We can set the diagonal to 0 and the sum of all entries to $K(K-1)$.
- Class priors, on the other hand, have $K-1$ degrees of freedom
  - For a given true class all misclassifications are assumed to have the same cost.
- This implies that if we want to respond to class distribution shift we may get away with a one-vs-rest perspective, but not if we're dealing with arbitrary changes in misclassification costs.

Note that now the decision rule is an arg-max over a class-weighted probability vector rather than a threshold on the predicted probability of a selected class.

Common sources of miscalibration

Overconfidence: a classifier is **worse** at separating classes than its scores suggest.

- Hence we need to *push predicted probabilities toward the centre*.

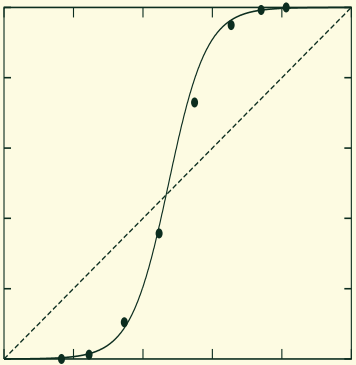
Underconfidence: a classifier separates classes **better** than its scores suggest.

- Hence we need to *pull predicted probabilities away from the centre*.

A classifier can be overconfident for one class and underconfident for the other, in which case all predicted probabilities need to be increased or decreased (in the relatively straightforward case of binary classification – more about multi-class later).

Underconfidence example

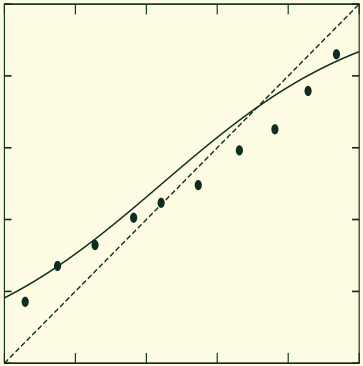
- Underconfidence typically gives **sigmoidal** distortions.
- To calibrate these means to *pull predicted probabilities away from the centre*.



Source: (Niculescu-Mizil and Caruana, 2005)

Overconfidence example

- Overconfidence is very common, particularly in deep neural networks. It is usually a consequence of over-counting evidence.
- It manifests itself through **inverse-sigmoidal** distortions
- Calibrating these means to *push predicted probabilities toward the centre*.



Source: (Niculescu-Mizil and Caruana, 2005)

A first look at some calibration techniques

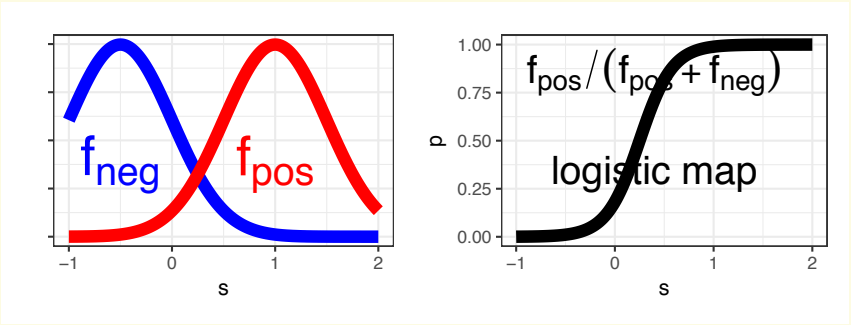
Parametric calibration involves modelling the score distributions within each class.

- **Platt scaling** = Logistic calibration; can be derived by assuming that the scores within both classes are normally distributed with the same variance (Platt, 2000).
- **Beta calibration** employs Beta distributions instead, to deal with scores already on a $[0, 1]$ scale (Kull et al., 2017).
- **Dirichlet calibration** for more than two classes (Kull et al., 2019).

Non-parametric calibration often ignores scores and employs ranks instead.

- E.g., **isotonic regression** = pool adjacent violators = ROC convex hull (Zadrozny and Elkan, 2001; Fawcett and Niculescu-Mizil, 2007).

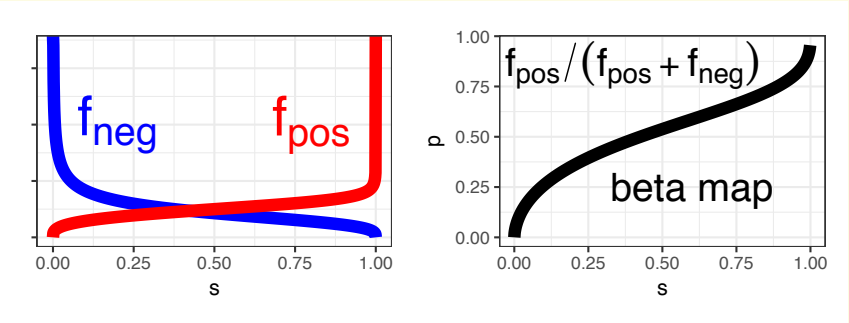
Platt scaling (aka logistic calibration) in a nutshell



$$p(s; w, m) = \frac{1}{1 + \exp(-w(s - m))}$$

$$w = (\mu_{pos} - \mu_{neg})/\sigma^2, m = (\mu_{pos} + \mu_{neg})/2$$

Beta calibration in a nutshell

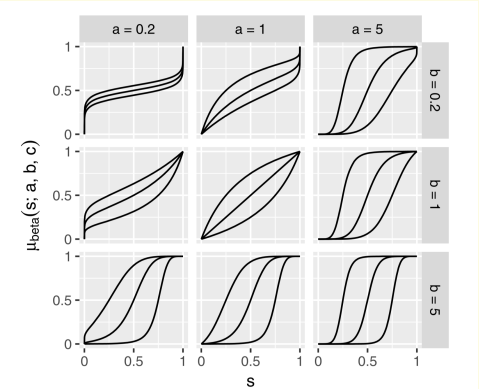


$$p(s; a, b, c) = \frac{1}{1 + \exp(-a \ln s - b \ln(1 - s) - c)}$$

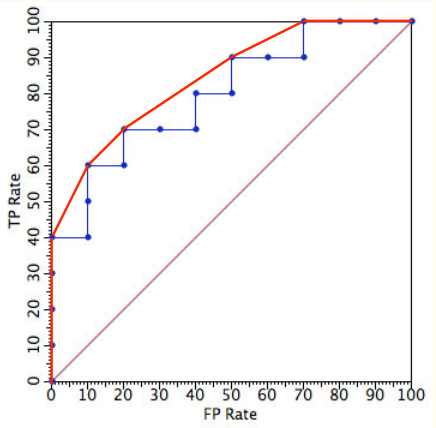
$$a = \alpha_{pos} - \alpha_{neg}, b = \beta_{neg} - \beta_{pos}$$

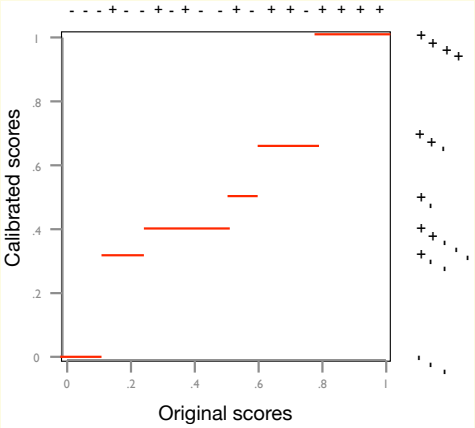
Beta calibration can deal with both under- and over-confidence

With Beta densities we can model a wide family of calibration maps, including the identity map for models that are already (nearly) calibrated.



Isotonic regression (aka ROC convex hull) in a nutshell





Source: (Flach, 2016)

What about multi-class calibration?

Similar to classification, some calibration methods are inherently multi-class but most are not.

This leads to (at least) three different ways of **defining** what it means to be fully multiclass-calibrated.

- Many recent papers use the (weak) notion of confidence calibration.

Evaluating multi-class calibration is in its full generality still an open problem.

Definitions of calibration for more than two classes

The following definitions of calibration are equivalent for binary classification but increasingly stronger for more than two classes:

Confidence calibration: only consider the highest predicted probability.

Class-wise calibration: only consider marginal probabilities.

Multi-class calibration: consider the entire vector of predicted probabilities.

Confidence calibration

This was proposed by (Guo et al., 2017), requiring that among all instances where the probability of **the most likely class** is predicted to be c, the expected accuracy is c. (We call this 'confidence calibration' to distinguish it from the stronger notions of calibration.)

Formally, a probabilistic classifier $\hat{\mathbf{p}} : \mathcal{X} \to \Delta_k$ is **confidence-calibrated**, if for any confidence level $c \in [0, 1]$, the actual proportion of the predicted class, *among all possible instances* $\mathbf{x}$ *being predicted the same class with the same confidence*, is equal to c:

$$P(Y = i \mid \hat{p}_i(\mathbf{x}) = c) = c \qquad \text{where } i = \operatorname*{argmax}_j \hat{p}_j(\mathbf{x}).$$

Class-wise calibration

Originally proposed by (Zadrozny and Elkan, 2002), this requires that all **one-vs-rest** probability estimators obtained from the original multi-class model are calibrated. This would allow to properly respond to **changes in class distribution**.

Formally, a probabilistic classifier $\hat{\mathbf{p}} : \mathcal{X} \to \Delta_k$ is **classwise-calibrated**, if for any class i and any predicted probability q_i for this class, the actual proportion of class i, *among all possible instances* $\mathbf{x}$ *getting the same prediction for that class*, is equal to q_i:

$$P(Y = i \mid \hat{p}_i(\mathbf{x}) = q_i) = q_i \qquad \text{for } \; i = 1, \ldots, k.$$

Multi-class calibration

This is the **strongest form of calibration** for multiple classes, subsuming the previous two definitions.

A probabilistic classifier $\hat{\mathbf{p}} : \mathcal{X} \to \Delta_k$ is **multiclass-calibrated** if
for any prediction vector $\mathbf{q} = (q_1, \ldots, q_k) \in \Delta_k$, the proportions of classes *among all possible instances* $\mathbf{x}$ *getting the same prediction vector* are equal to $\mathbf{q}$:

$$P(Y = i \mid \hat{\mathbf{p}}(\mathbf{x}) = \mathbf{q}) = q_i \qquad \text{for } i = 1, \ldots, k.$$

This would allow to properly respond to **arbitrary changes in misclassification costs**.

Reminder: binning needed

For practical purposes, the conditions in these definitions need to be relaxed. This is where **binning** comes in.

Once we have the bins, we can draw a **reliability diagram** as in the two-class case. For class-wise calibration, we can show per-class reliability diagrams or a single averaged one.

One way to assess the degree of calibration is by means of the **gaps** in the reliability diagram.

Important points to remember so far

Only well-calibrated probability estimates are worthy to be called probabilities:
otherwise they are just scores that happen to be in the $[0, 1]$ range.

Binning will play a role at some point:
instance-based metrics such as Brier score or log-loss (see later) always measure calibration **plus something else**: decomposing this requires binning.

In multi-class settings, think carefully about which form of calibration you need:
e.g., confidence-calibration is too weak in a cost-sensitive setting.

I will now talk about...

How to measure calibration

- Expected Calibration Error (ECE)
- Proper scoring rules

Expected Calibration Error (ECE)

- As seen in the previous section, each notion of calibration is related to a reliability diagram
  - This can be used to visualise miscalibration on binned scores
- We will now see how these bins can be used to measure miscalibration

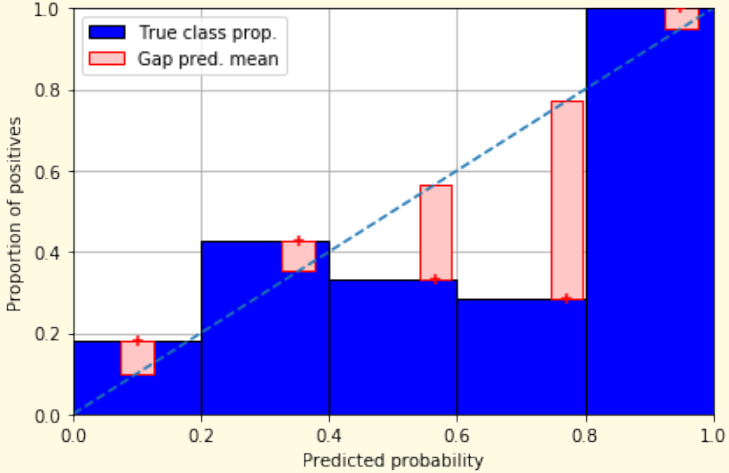
Binary-ECE

- We define the expected binary calibration error binary−ECE (Naeini et al., 2015) as the average gap across all bins in a reliability diagram, weighted by the number of instances in each bin:

$$\text{binary}-\text{ECE} = \sum_{i=1}^{M} \frac{|B_i|}{N} |\bar{y}(B_i) - \bar{p}(B_i)|,$$

- Where M and N are the numbers of bins and instances, respectively, B_i is the i-th probability bin, $|B_i|$ denotes the size of the bin, and $\bar{p}(B_i)$ and $\bar{y}(B_i)$ denote the average predicted probability and the proportion of positives in bin B_i

ECE aggregates the red bars in a reliability diagram



Binary-ECE in numbers

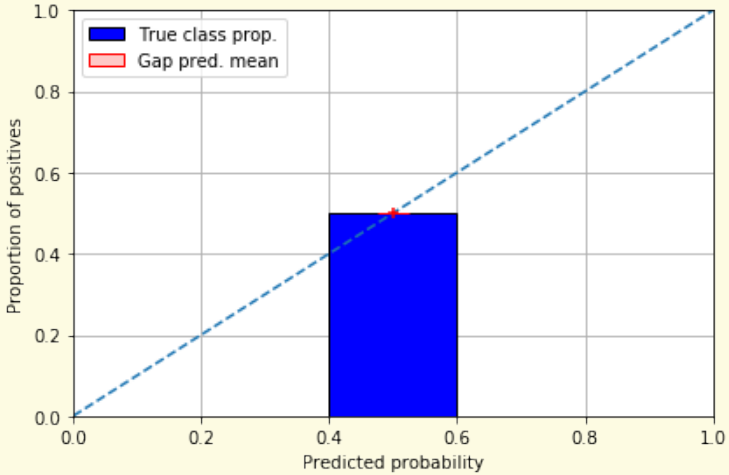
B_i	$\bar{p}(B_i)$	$\bar{y}(B_i)$	$\lvert B_i \rvert$
B_1	0.10	0.18	11
B_2	0.35	0.43	7
B_3	0.57	0.33	3
B_4	0.77	0.29	7
B_5	0.95	1.00	2

$$\text{binary}-\text{ECE} = \sum_{i=1}^{M} \frac{|B_i|}{N} |\bar{y}(B_i) - \bar{p}(B_i)|$$

$$\text{binary}-\text{ECE} = \frac{11 \cdot 0.08 + 7 \cdot 0.08 + 3 \cdot 0.24 + 7 \cdot 0.48 + 2 \cdot 0.05}{30}$$

$$\text{binary}-\text{ECE} = 0.1873$$

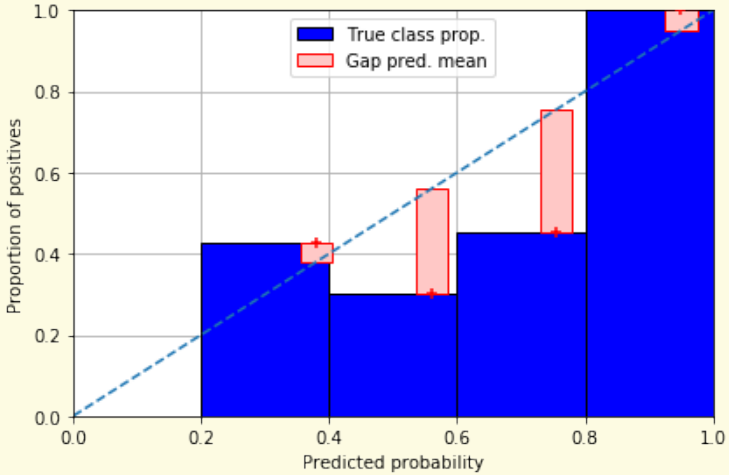
NB. Always predicting the class prior trivially optimises ECE!



Confidence-ECE

- Confidence-ECE (Guo et al., 2017) was the first attempt at an ECE measure for multi-class problems with $K > 2$ classes
- Here, confidence means the probability given to the winning class, i.e. the highest value in the predicted probability vector
- We calculate the expected confidence calibration error as the binary-ECE of the binned confidence values

Confidence-ECE reliability diagram



Confidence-ECE in numbers

B_i	$\bar{p}(B_i)$	$\bar{y}(B_i)$	$\lvert B_i \rvert$
B_1			0
B_2	0.38	0.43	7
B_3	0.56	0.30	10
B_4	0.75	0.45	11
B_5	0.95	1.00	2

$$\text{confidence}-\text{ECE} = \sum_{i=1}^{M} \frac{|B_i|}{N} |\bar{y}(B_i) - \bar{p}(B_i)|$$

$$\text{confidence}-\text{ECE} = \frac{0 + 7 \cdot 0.05 + 10 \cdot 0.26 + 11 \cdot 0.3 + 2 \cdot 0.05}{30}$$

$$\text{confidence}-\text{ECE} = 0.2117$$

Classwise-ECE

- Confidence calibration only cares about the winning class.
- To measure miscalibration for all classes, we can take the average binary-ECE across all classes (one-vs-rest).

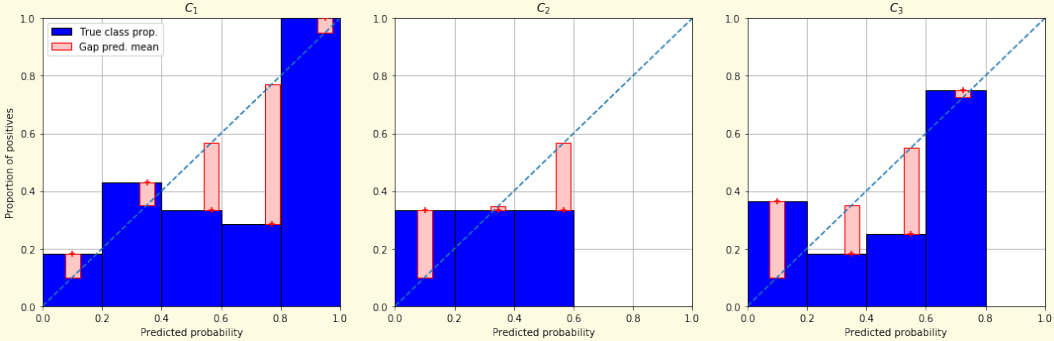
Classwise-ECE

- Formally, classwise−ECE is defined as the average gap across all classwise-reliability diagrams, weighted by the number of instances in each bin:

$$\text{classwise}-\text{ECE} = \frac{1}{K}\sum_{j=1}^{K}\sum_{i=1}^{M}\frac{|B_{i,j}|}{N}|\bar{y}_j(B_{i,j}) - \bar{p}_j(B_{i,j})|,$$

- Where $B_{i,j}$ is the i-th bin of the j-th class, $|B_{i,j}|$ denotes the size of the bin, and $\bar{p}_j(B_{i,j})$ and $\bar{y}_j(B_{i,j})$ denote the average prediction of class j probability and the actual proportion of class j in the bin $B_{i,j}$

Each class now has its own reliability diagram



What about multiclass-ECE?

- True multiclass-ECE is still an open problem
- With large numbers of classes, the number of bins can be prohibitively high
  - Most bins would be empty
- Therefore, we turn to **proper scoring rules**

Proper scoring rules

- These are loss measures (ϕ) that always prefer Bayes-optimal classifiers over other classifiers.
- This means that for the true $P(\mathbf{X}, Y)$, $\mathbf{x} \in \mathcal{X}$, the following is satisfied:

$$\mathbb{E}_{y \sim P(Y|\mathbf{X}=\mathbf{x})}\Big[\phi\big(\mathbf{q}, y\big)\Big] \geq \mathbb{E}_{y \sim P(Y|\mathbf{X}=\mathbf{x})}\Big[\phi\big(P(Y \mid \mathbf{X}=\mathbf{x}), y\big)\Big]$$

- The left-hand side is equal to the right-hand side if and only if $\mathbf{q} = P(Y \mid \mathbf{X}=\mathbf{x})$
- $P(Y \mid \mathbf{X}=\mathbf{x})$ is a vector with elements $P(Y=j \mid \mathbf{X}=\mathbf{x})$

Brier score (aka quadratic error)

$$\phi_{\text{BS}}\Big(\mathbf{Q}, \mathbf{y}\Big) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{K}\Big(\mathbb{I}(y_n = j) - q_{n,j}\Big)^2$$

- We can easily see that this value is not minimised by constantly predicting the class distribution, as in ECE

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \qquad \phi_{\text{BS}}\Big(\mathbf{Q}, \mathbf{y}\Big) = \frac{(1-0.5)^2 + (0-0.5)^2 + (0-0.5)^2 + (1-0.5)^2}{2}$$

$$\mathbf{y} = [1, 2] \qquad \phi_{\text{BS}}\Big(\mathbf{Q}, \mathbf{y}\Big) = 0.5$$

Log-loss (aka cross-entropy)

$$\phi_{\text{LL}}\Big(\mathbf{Q}, \mathbf{y}\Big) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{K}\mathbb{I}(y_n = j)\cdot \ln(q_{n,j})$$

- Frequently used as the training loss of optimisation-based learning methods, such as neural networks
- Only considers the probability given to the true class!

$$\phi_{\text{LL}}\Big(\mathbf{Q}, \mathbf{y}\Big) = -\frac{(1\cdot\ln(0.5) + 0\cdot\ln(0.5) + 0\cdot\ln(0.5) + 1\cdot\ln(0.5))}{2}$$

$$\phi_{\text{LL}}\Big(\mathbf{Q}, \mathbf{y}\Big) = -\ln 0.5 = 0.6931$$

Why are these losses proper?

$$\phi(q, y) = y\phi(q, 1) + (1 - y)\phi(q, 0)$$
$$\mathbb{E}_{y\sim p}\phi(q, y) = p\phi(q, 1) + (1 - p)\phi(q, 0)$$
$$\frac{\partial}{\partial q}\mathbb{E}_{y\sim p}\phi(q, y) = p\phi'(q, 1) + (1 - p)\phi'(q, 0)$$

$\phi(q,1)$	$\phi'(q,1)$	$\phi'(q,0)$	$p\phi'(q,1) + (1-p)\phi'(q,0)$
$(1-q)^2$	$-2(1-q)$	$2q$	$-2p(1-q) + 2(1-p)q = 2(q-p)$
$-\ln q$	$-1/q$	$1/(1-q)$	$-p/q + (1-p)/(1-q) = (q-p)/(q(1-q))$

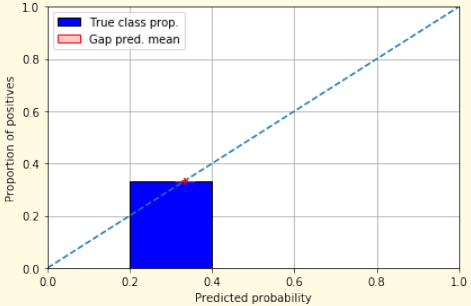
So $q = p$ is the unique minimiser for both Brier score and log-loss.
In contrast, MAE is minimised by $q = 1$ ($q = 0$) whenever $p > 0.5$ ($p < 0.5$).

An evaluation trade-off

- As mentioned before, a model that always outputs the class proportion will have a perfect ECE of 0, but its log-loss is not 0 (here, it's $-\frac{2}{3}\ln\frac{2}{3} - \frac{1}{3}\ln\frac{1}{3} = 0.6365$)

$$\mathbf{Q} = \begin{bmatrix} 2/3 & 1/3 \\ 2/3 & 1/3 \\ \vdots & \vdots \\ 2/3 & 1/3 \end{bmatrix}$$

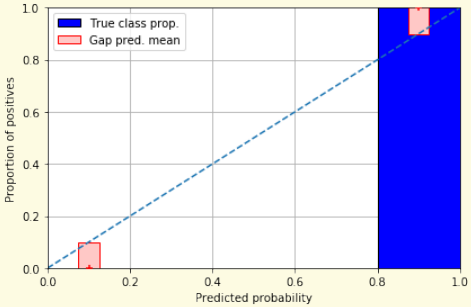
$$\mathbf{y} = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0]$$



A better model can have a worse ECE

- What happens if our model gives 0.9 probability to the instances' true classes?

$$\text{accuracy} = 1$$
$$\text{ECE} = 0.1$$
$$\text{log-loss} = 0.1054$$



- ECE increased as the more discriminative model was less well calibrated, while log-loss decreased.

Proper scoring rules measure more than calibration

- An intuitive way to decompose proper scoring rules is into refinement and calibration losses: $\mathbb{E}[\phi] = \text{RL} + \text{CL}$.
  - **Refinement loss** is the loss due to producing the same probability for instances from different classes – this loss was maximum for the first model and zero for the second.
  - **Calibration loss** is the loss due to the difference between the probabilities predicted by the model and the proportion of positives among instances with the same prediction – this loss was zero for the first model but slightly higher for the second.
- Since we don't usually know the real score distribution, we would need to once again rely on **binning** if we wanted to actually estimate refinement and calibration losses – see cost curves later.
- Note that the proper score of the Bayes-optimal classifier is not necessarily zero.
  - This is due to irreducible loss, which is zero only if the data is separable (Kull and Flach, 2015).

Calibration metrics: summary

- There are various ways to visualise and quantify calibration
- ECE measures aim at producing an aggregate measure of the visual information provided in reliability diagrams
  - Thus, their optimisation is **not** guaranteed to produce desirable classifiers
- Proper scoring rules measure different aspects of probability correctness
  - They have been used as training losses in classifier training for a while

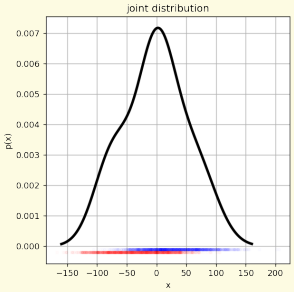
I will now talk about...

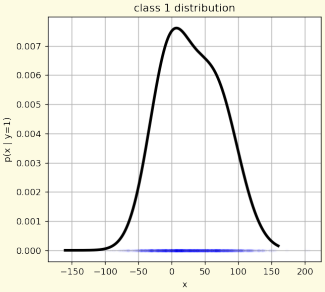
Post-hoc calibrators in action

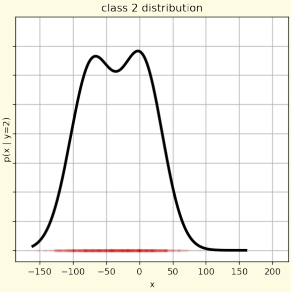
- A toy dataset
- Binary calibrators
- Multi-class calibrators

The synthetic dataset

We use a simple univariate two-class dataset for illustrative purposes.



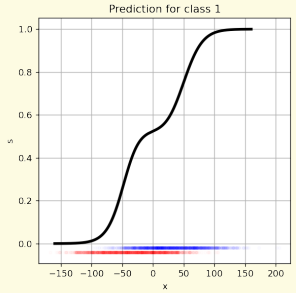




The synthetic dataset (discriminative view)

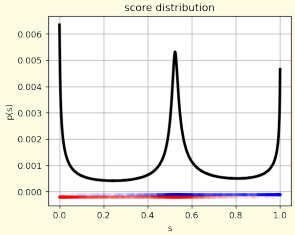
As we know the data distribution, we can calculate the Bayes-optimal scoring model:

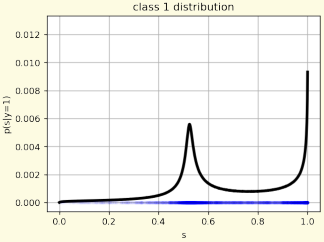
$$s(x) = P(Y = 1|X = x)$$

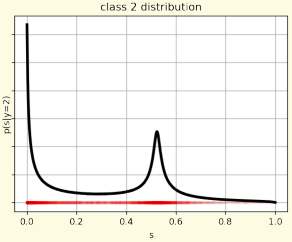


The distribution of predicted scores

We can further calculate the overall and conditional distributions of scores:

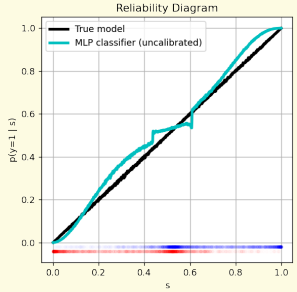


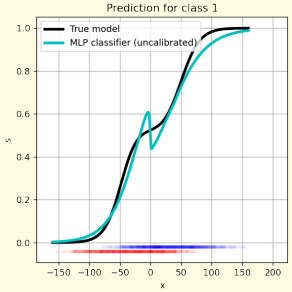




An uncalibrated classifier

Now we take a sample and train a multi-layer perceptron (MLP).





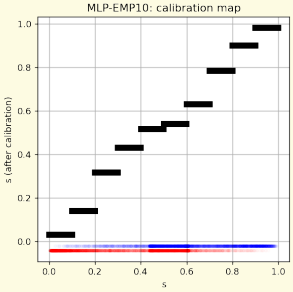
The ideal post-hoc calibrator would exactly model the distortions in the reliability diagram.

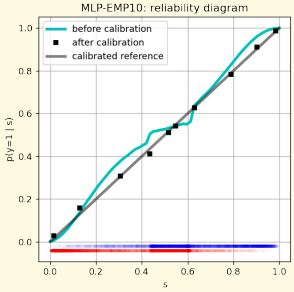
Binary calibrators

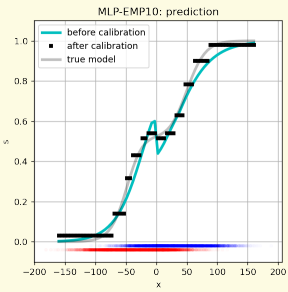
- J. Platt. Probabilities for SV Machines.
  In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large-Margin Classifiers*, pages 61—-74. MIT Press, 2000
- M. P. Naeini and G. F. Cooper. Binary Classifier Calibration Using an Ensemble of Near Isotonic Regression Models.
  In *IEEE 16th International Conference on Data Mining (ICDM)*, pages 360–369. Institute of Electrical and Electronics Engineers (IEEE), 2016
- M. Kull, T. M. Silva Filho, and P. Flach. Beyond Sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017

Empirical Binning

While being simple and effective, binning approaches can only give discrete outputs.

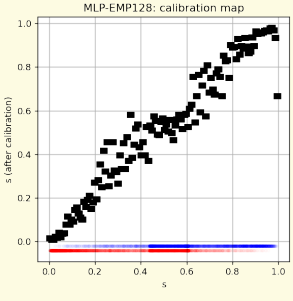


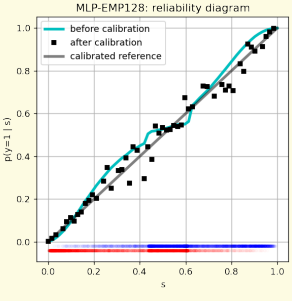


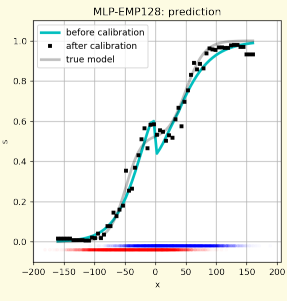


Empirical Binning

A suitable number of bins / binning algorithm is important to get good results.

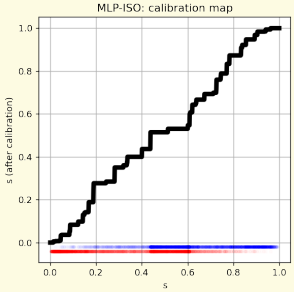


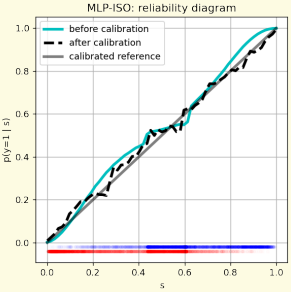


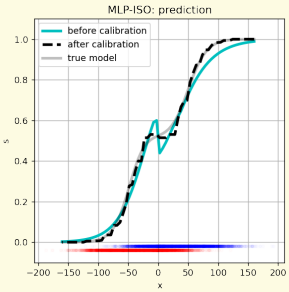


Isotonic Regression

With the ROC-convex hull method, isotonic regression can give good calibration performance with automatic binning and interpolation.

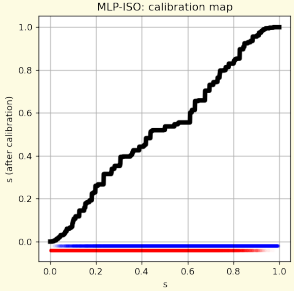


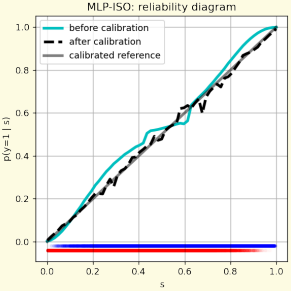


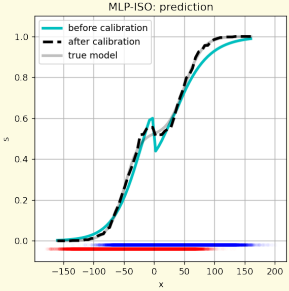


Isotonic Regression

More data points are beneficial for non-parametric methods such as isotonic regression.

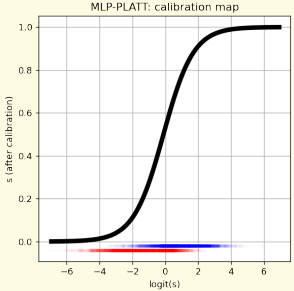


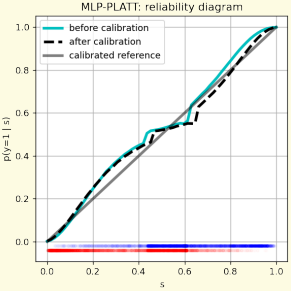


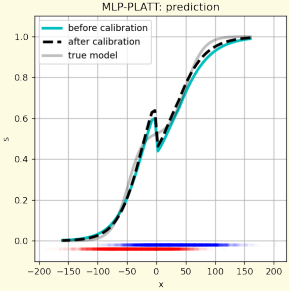


Platt scaling

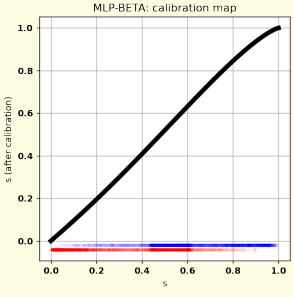
This is like a parametrised softmax in the final layer of the MLP, obtained by performing logistic regression on the **logits**.

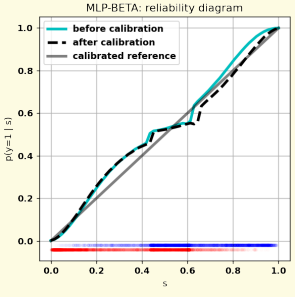


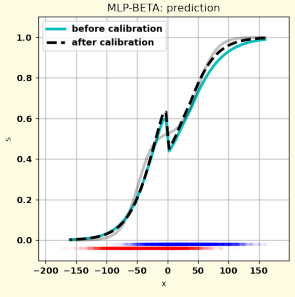




Beta Calibration



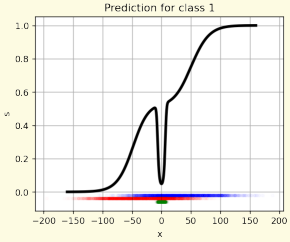


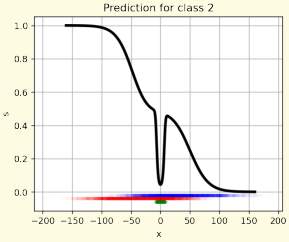


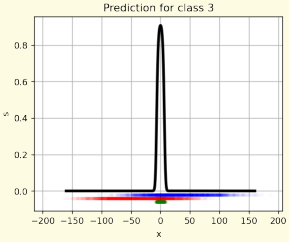
Multi-class calibrators

- C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks.
  In *34th International Conference on Machine Learning*, pages 1321–1330, Sydney, Australia, 2017
- M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration.
  In *Advances in Neural Information Processing Systems (NIPS'19)*, pages 12316–12326, 2019

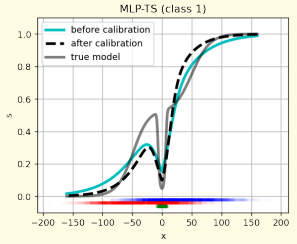
Let’s add a (hard) third class

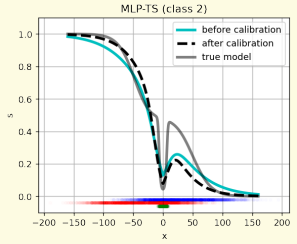


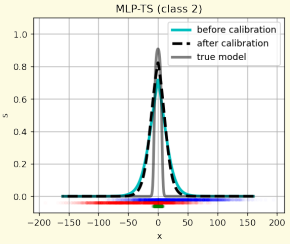


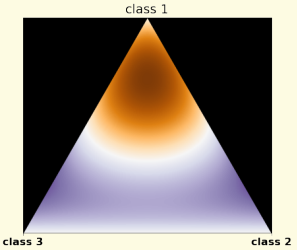


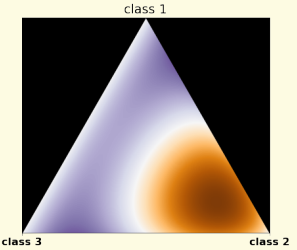
Temperature Scaling

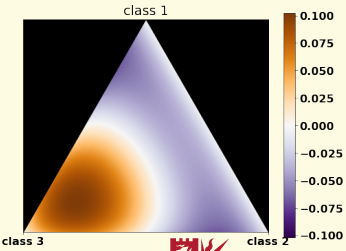




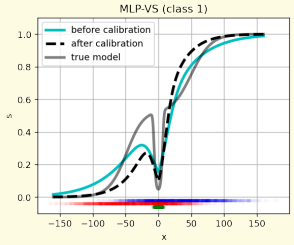


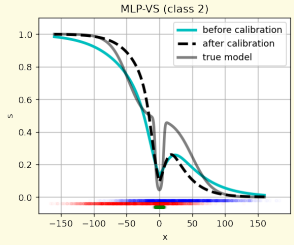


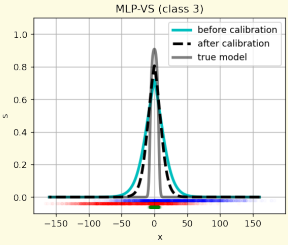


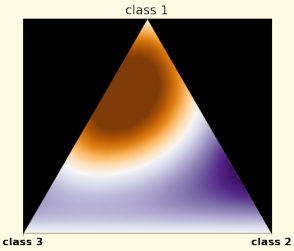


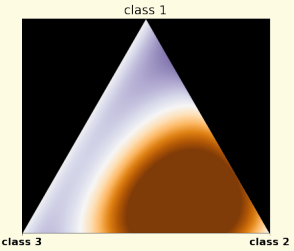
Vector Scaling

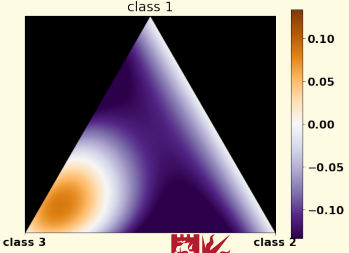




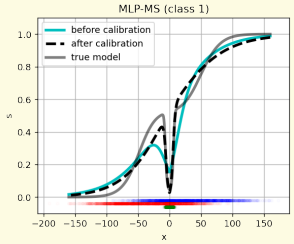


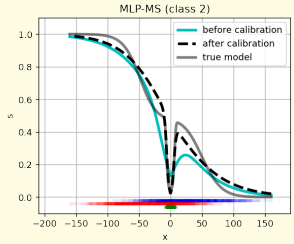


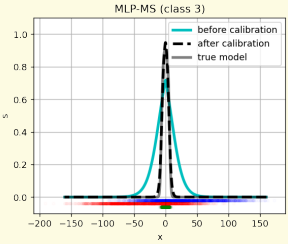


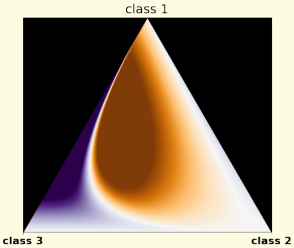


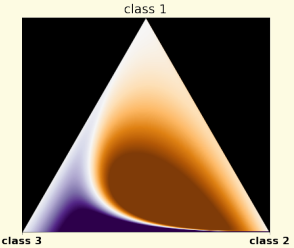
Matrix Scaling

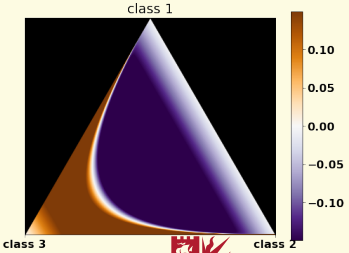




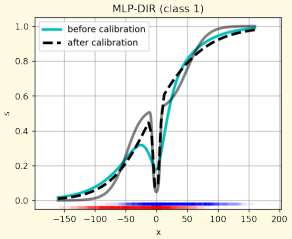


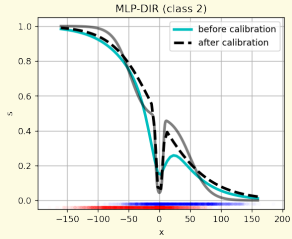


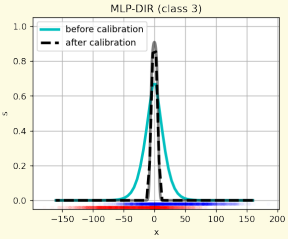


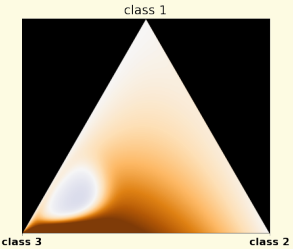


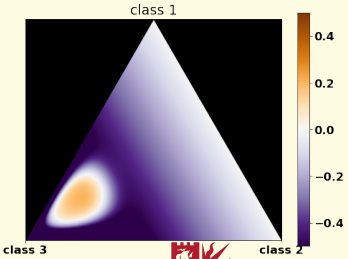
Dirichlet Calibration











I will now talk about...

Some advanced topics

- Brier curves
- Calibration for F-score
- Calibration for regression

An alternative view of calibration

So far we have framed calibration as aiming to get as close as possible to the Bayes-optimal classifier, predicting the 'true' $P(y|x)$.

An alternative view starts from the observation that the output of a calibrated classifier allows to calculate the cost ratio (or class prior) under which an instance is on the decision boundary.

This view leads to a useful visualisation of calibration by means of **Brier curves**. It also opens the way to calibrating for alternative classification performance measures, in particular **F-score**.

Brier curves I

As we have seen before, relative misclassification costs are given by the cost parameter $c = \frac{c_{\text{FP}}}{c_{\text{FP}}+c_{\text{FN}}} \in [0,1]$.

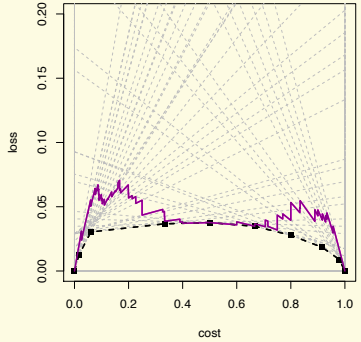
Decision theory shows that the optimal decision threshold on well-calibrated probabilities is c. Let π denote the proportion of positives.

The normalised loss at threshold c is then

$$L(c) = 2c(1-\pi)fpr(c) + 2(1-c)\pi\,(1-tpr(c))$$

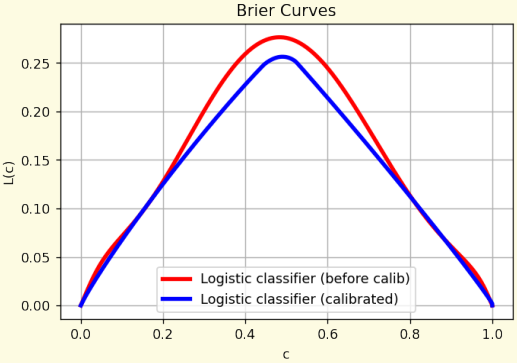
The **Brier curve** plots $L(c)$ against c (Hernández-Orallo et al., 2011).
It is a particular kind of cost curve (Drummond and Holte, 2006).

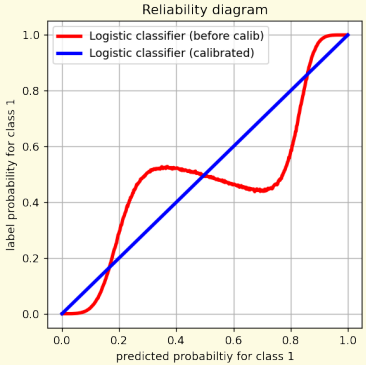
Brier curves II



- Cost lines arise from fixed operating points (thresholds).
- The area under the Brier curve (expected loss over uniform c) is the Brier score.
- The lower envelope of the cost lines would be achieved by perfect calibration.

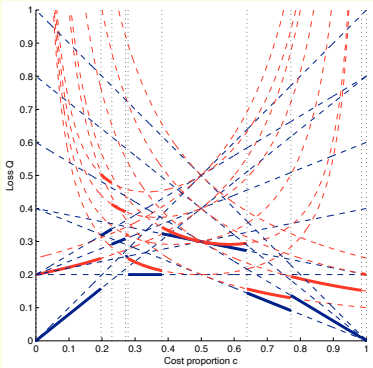
Brier curves III

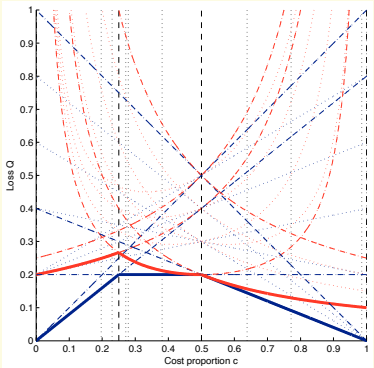




- The area under the red curve is the uncalibrated model's Brier score;
- the area under the blue curve is the refinement loss;
- the area between the two curves is the calibration loss of the uncalibrated model.

Brier curves IV





We can obtain similar curves for log-loss and other proper scoring rules (Flach, 2015).

Reinterpreting the output of a calibrated classifier

A calibrated score of $p = r/(r+1)$ tells us that this instance's predicted class wouldn't affect performance if negatives are $r = p/(1-p)$ times more important than positives.

So each calibrated score p has an associated **weighted accuracy measure** $acc_p = 2p \cdot (1-\pi)tnr + 2(1-p) \cdot \pi tpr$ for which instances with that score are on the decision boundary.

Q: What changes if we are interested in F-score rather than accuracy?

Calibrating for F-score

F_β is a weighted harmonic mean of precision and recall:

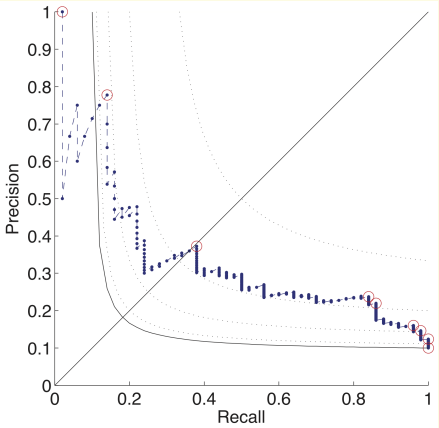
$$F_\beta \triangleq \frac{1}{\frac{1}{1+\beta^2}\Big/ prec + \frac{\beta^2}{1+\beta^2}\Big/ rec} = \frac{(1+\beta^2)TP}{(1+\beta^2)TP + FP + \beta^2 FN}$$

It is more convenient to have a $[0,1]$ weight $d = 1/(1+\beta^2)$:

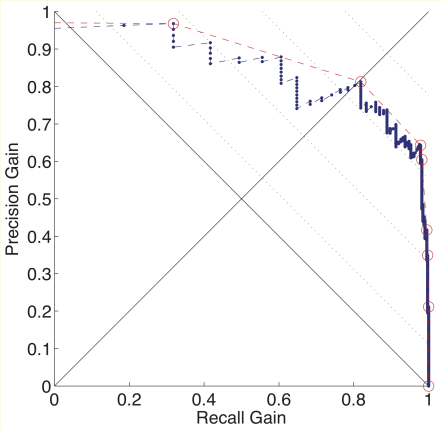
$$F_d \triangleq \frac{1}{d/prec + (1-d)/rec} = \frac{TP}{TP + d \cdot FP + (1-d) \cdot FN}$$

Q: Can we construct a calibration procedure that produces the value of d for which an instance is on the F_d decision boundary?

PRG curves to the rescue (Flach and Kull, 2015)

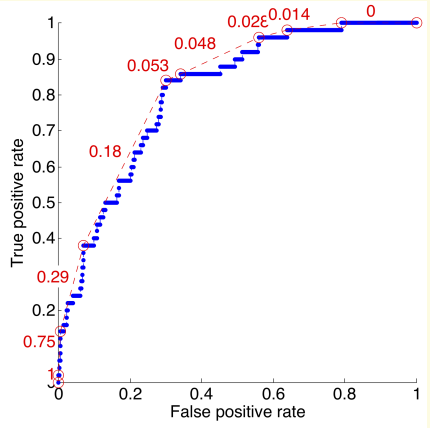


Precision-recall curve

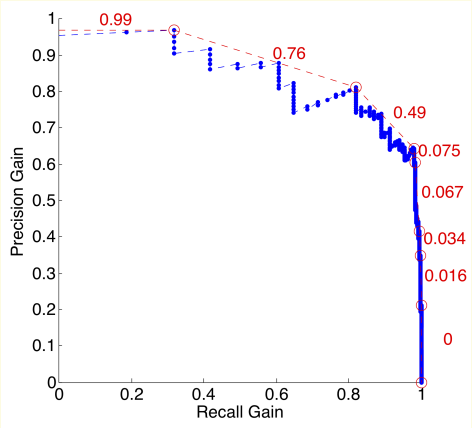


Linearised PRG curve

Isotonic regression applied to PRG curve



ROC curve with accuracy-calibrated scores

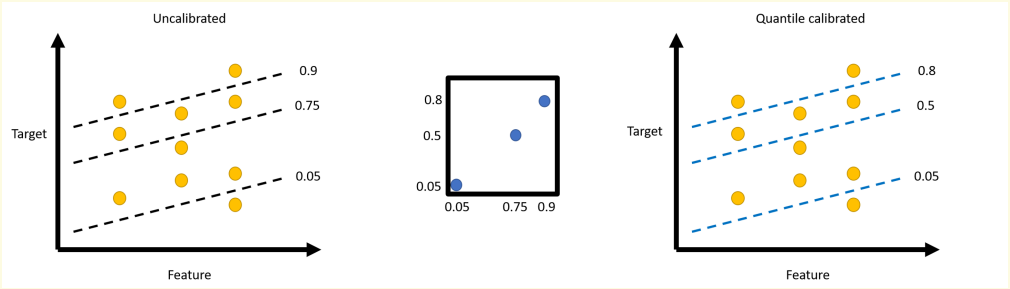


PRG curve with F_d-calibrated scores

Quantile-Calibrated Regression

A quantile regression model $g : \mathbb{X} \times [0, 1] \to \mathbb{R}$ is **quantile-calibrated** if and only if

$$P\Big(Y \leq g(\mathbf{X}, \tau)\Big) = \tau \qquad \text{for } \forall \tau \in [0, 1]$$



Distribution-Calibrated Regression

A regression model (aka conditional density estimator)

$$f : \mathbb{X} \to \{s \mid s : \mathbb{R} \to [0, \infty), \int s(y)dy = 1\}$$

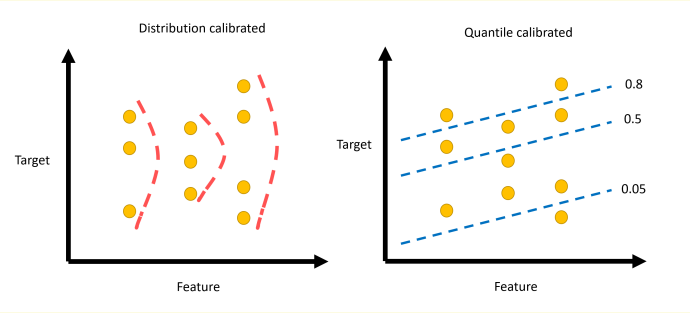
is **distribution-calibrated** if and only if

$$P\Big(Y = y \mid f(\mathbf{X}) = s\Big) = s(y)$$

Being distribution-calibrated is a sufficient but not necessary condition for being quantile-calibrated (Song et al., 2019, Theorem 1).

Distribution-Calibration vs. Quantile-Calibration

Quantile-calibrated regressors model the global quantiles, while distribution-calibrated regressors model conditional distributions.



I will now talk about...

Concluding remarks

- The myth of the decision boundary
- The long history of classifier calibration
- Calibration is an art as well as a science

The myth of the decision boundary

Resist thinking in terms of decision boundaries.

- That is like describing Mount Everest with a single contour line, halfway up.
- To faithfully characterise the mountain you need many contour lines at different elevations.

The **only case** in which a single decision boundary is useful is when the class prior (and misclassification costs, if any) won't change after training.

- In that case you don't need to calibrate the entire probability range, only the decision threshold.

The long history of classifier calibration

Contrary what recent machine learning literature may lead you to believe, **calibration research predates machine learning** and has been studied for three-quarters of a century.

- Make sure you take full advantage of that history, and don't just follow the citations in the latest NeurIPS paper on DNN calibration.

Recent proposals such as confidence calibration, temperature scaling and expected calibration error have their uses for sure, but they are not the only options and may not be suitable for your particular context or application.

- Which leads me to...

Calibration is an art as well as a science

Many practical questions don't have *a priori* answers:

- Which calibration method?
- How many bins?
- What evaluation metric?
- Do I use a validation set? How large?
- ...

T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach.

Classifier calibration: a survey on how to assess and improve predicted class probabilities.
Machine Learning, 112(9):3211–3260, 2023

References I

G. W. Brier. Verification of forecasts expressed in terms of probabilities. *Monthly Weather Review*, 78(1):1–3, 1950.

C. Drummond and R. C. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95–130, 2006.

T. Fawcett and A. Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 68(1):97–106, 2007.

P. Flach. Classification in context: Adapting to changes in class and cost distribution. In *First international workshop on learning over multiple contexts at ECML-PKDD'14*, 2014. URL `http://dmip.webs.upv.es/LMCE2014/Papers/lmce2014_submission_18.pdf`.

References II

P. Flach. Cost-Sensitive Classification Meets Proper Scoring Rules. In *Second international workshop on learning over multiple contexts at ECML-PKDD'15*, 2015. URL `http://dmip.webs.upv.es/LMCE2015/Papers/LMCE_2015_submission_5.pdf`.

P. Flach. ROC analysis. In *Encyclopedia of Machine Learning and Data Mining*. Springer, 2016.

P. Flach and M. Kull. Precision-Recall-Gain Curves: PR Analysis Done Right. In *Advances in Neural Information Processing Systems (NIPS'15)*, pages 838–846, 2015.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In *34th International Conference on Machine Learning*, pages 1321–1330, Sydney, Australia, 2017.

References III

J. Hernández-Orallo, P. Flach, and C. Ferri. Brier Curves: A New Cost-Based Visualisation of Classifier Performance. In *28th International Conference on Machine Learning (ICML'11)*, pages 585—-592, 2011.

M. Kull and P. Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD'15)*, volume 9284, pages 68–85. Springer Verlag, 2015.

M. Kull, T. M. Silva Filho, and P. Flach. Beyond Sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017.

References IV

M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems (NIPS'19)*, pages 12316–12326, 2019.

M. P. Naeini and G. F. Cooper. Binary Classifier Calibration Using an Ensemble of Near Isotonic Regression Models. In *IEEE 16th International Conference on Data Mining (ICDM)*, pages 360–369. Institute of Electrical and Electronics Engineers (IEEE), 2016.

P. Naeini, G. F. Cooper, and M. Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *29th AAAI Conference on Artificial Intelligence*, 2015.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *22nd International Conference on Machine Learning (ICML'05)*, pages 625–632, New York, New York, USA, 2005. ACM Press.

References V

J. Platt. Probabilities for SV Machines. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large-Margin Classifiers*, pages 61—-74. MIT Press, 2000.

T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112(9):3211–3260, 2023.

H. Song, T. Diethe, M. Kull, and P. Flach. Distribution calibration for regression. volume 97 of *Proceedings of Machine Learning Research*, pages 5897–5906, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/song19a.html`.

B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *18th International Conference on Machine Learning (ICML'01)*, pages 609—-616, 2001.

References VI

B. Zadrozny and C. Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *8th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, pages 694—-699, New York, New York, USA, 2002. ACM Press.

The Why and How of Classifier Calibration

Peter Flach, with slides contributed by *Telmo Silvo Filha* and *Hao Song*, and prepared in collaboration with *Miquel Perello Nieto*, *Meelis Kull* and *Raul Santos-Rodriguez*

`classifier-calibration.github.io/`