Calibration: What and Why

Classifier Calibration Tutorial
ECML PKDD 2020

Prof. Peter Flach
Peter.Flach@bristol.ac.uk
classifier-calibration.github.io/

Departamento de ESTATÍSTICA
University of BRISTOL
UNIVERSITY OF TARTU

Table of Contents

Table of Contents

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:

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

and to multi-class classification:

- 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 (observed) probabilities.

Q: What does 'x% of the time' mean?

Forecasting example

Let's consider a small toy example:

- Two predictions of '10% chance of rain' were both followed by 'no rain'.
- Two predictions of '40% chance of rain' were once followed by 'no rain', and once by 'rain'.
- Three predictions of '70% chance of rain' were once followed by 'no rain', and twice by 'rain'.
- One prediction of '90% chance of rain' was followed by 'rain'.

Q: Is this forecaster well-calibrated?

Over- and under-estimates

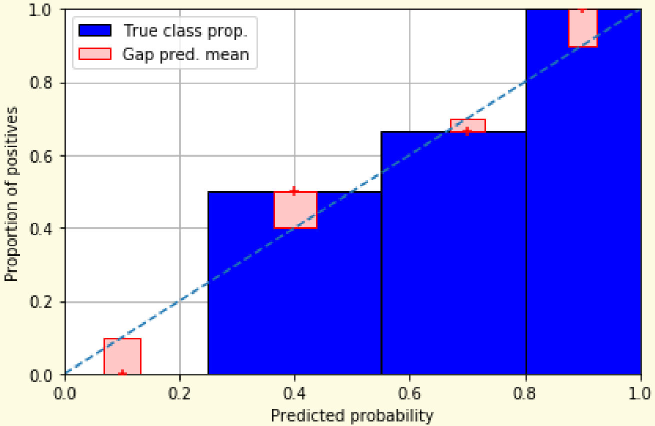
	$\hat{p}$	y
0	0.1	0
1	0.1	0
2	0.4	0
3	0.4	1
4	0.7	0
5	0.7	1
6	0.7	1
7	0.9	1

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 ($\bar{y} = 1/1 = 100\%$).

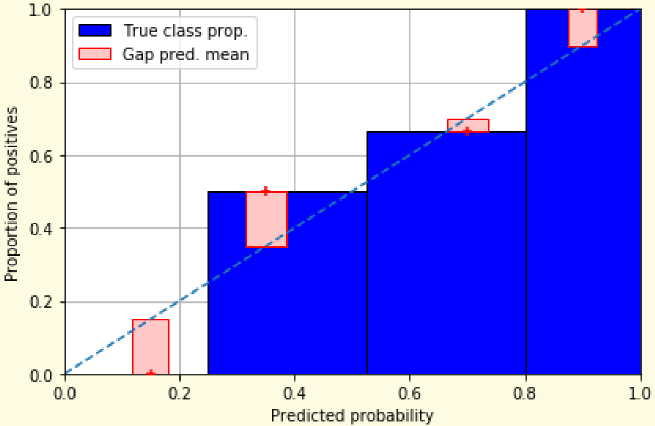
Visualising forecasts: the reliability diagram

	$\hat{p}$	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



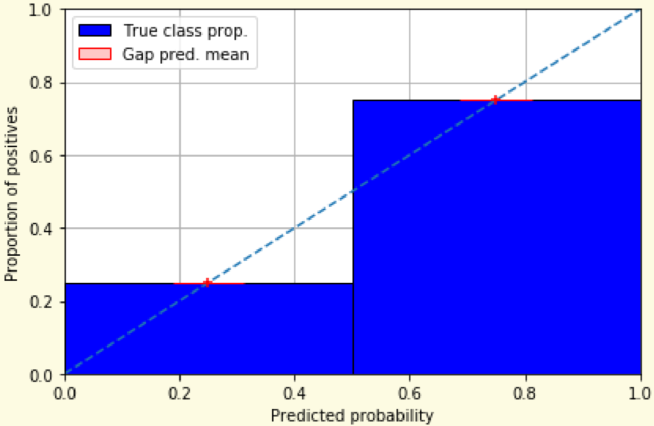
Changing the numbers slightly

	$\hat{p}$	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



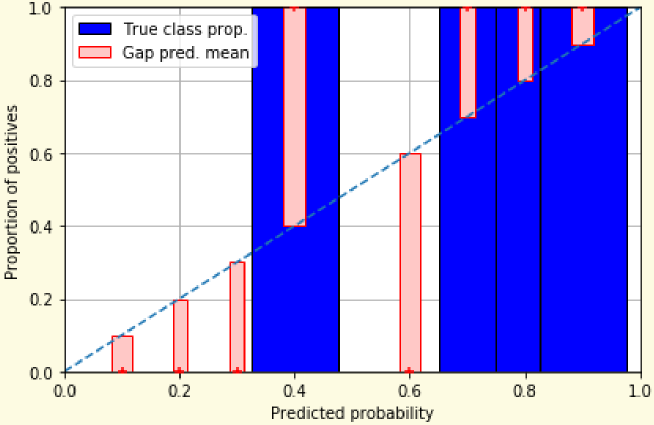
Or should we group forecasts differently?

	$\hat{p}$	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



Or not at all?

	$\hat{p}$	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 are we interested in calibration?

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 account for different 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?

Optimal decisions 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:

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

Optimal decisions 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, which gives

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

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!

Optimal decisions III

Without loss of generality we can set the cost of true positives and true negatives to zero; $c = \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}}+c_{\mathrm{FN}}}$ 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 in the obvious way.

Common sources of miscalibration

Underconfidence: a classifier thinks it's **worse** at separating classes than it actually is.

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

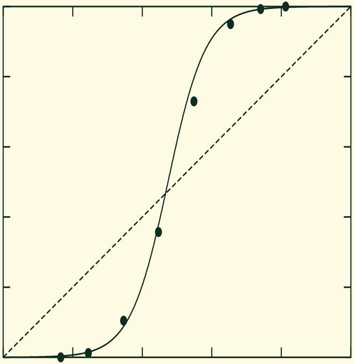
Overconfidence: a classifier thinks it's **better** at separating classes than it actually is.

- Hence we need to *push predicted probabilities toward 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.

Underconfidence example

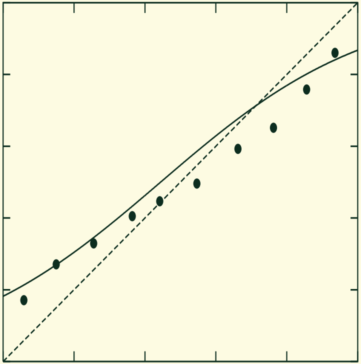
- Underconfidence typically gives **sigmoidal** distortions.
- To calibrate these means to *pull predicted probabilities away from the centre*.



Source: (Niculescu-Mizil and Caruana, 2005)

Overconfidence example

- Overconfidence is very common, and usually a consequence of over-counting evidence.
- Here, distortions are **inverse-sigmoidal**
- 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.

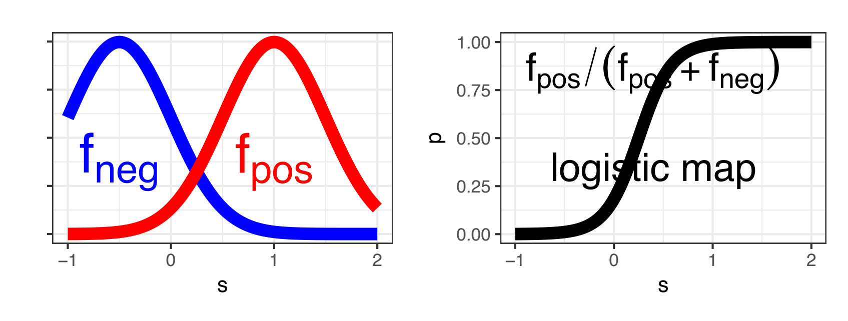
- **Platt scaling** = Logistic calibration can be derived by assuming that the scores within both classes are normally distributed with the same variance (Platt, 2000).
- **Beta calibration** employs Beta distributions instead, to deal with scores already on a $[0, 1]$ scale (Kull et al., 2017).
- **Dirichlet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

These techniques will be more fully discussed in Part III of the tutorial.

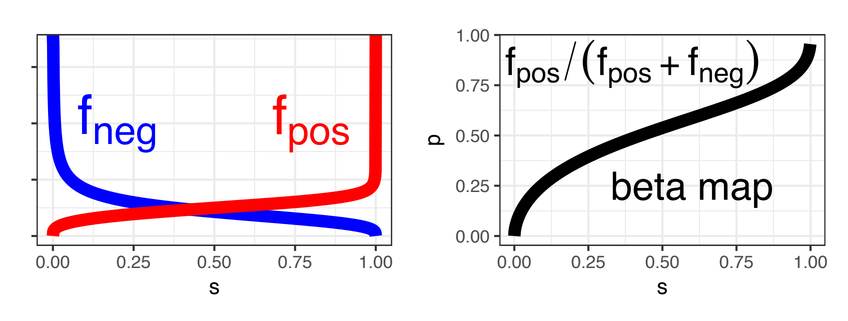
Platt scaling



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

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

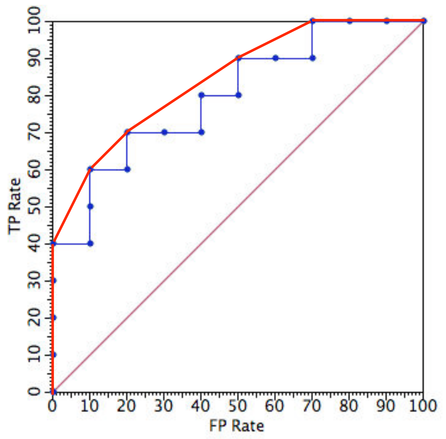
Beta calibration

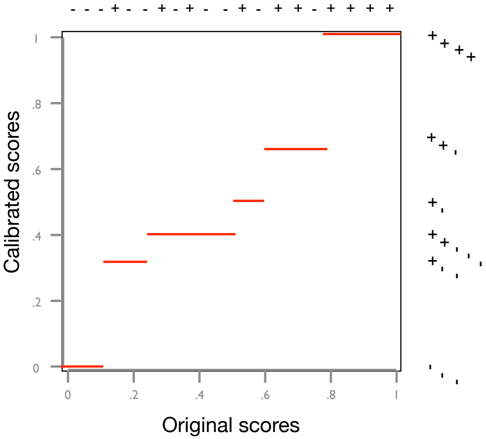


$$p(s; a, b, c) = \frac{1}{1 + \exp(-a \ln s - b \ln(1 - s) - c)}$$

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

Isotonic regression





Source: (Flach, 2016)

Table of Contents

What's so special about multi-class calibration?

Similar to classification, some 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 this class with confidence c, 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 multiclass model are calibrated.

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 $\hat{p}_i(\mathbf{x}) = q_i$, 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 $\hat{\mathbf{p}}(\mathbf{x}) = \mathbf{q}$ are equal to the prediction vector $\mathbf{q}$:

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

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.

The degree of calibration is assessed using the **gaps** in the reliability diagram. All of this will be elaborated in the next part of the tutorial.

Table of Contents

Important points to remember

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 be required in some form:

instance-based probability evaluation metrics such as Brier score or log-loss always measure calibration **plus something else**.

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.

What happens next

14.45 - Telmo Silva-Filho: Evaluation metrics and proper scoring rules

Expected/maximum calibration error; proper scoring rules; hypothesis test for calibration

15.30 - Break and preparation for hands-on session

15.50 - Hao Song: Calibrators

Binary approaches; multi-class approaches; regularisation and Bayesian treatments; implementation

16.50 - Miquel Perello-Nieto: Hands-on session

17.30 - Peter Flach, Hao Song: Advanced topics and conclusion

Cost curves; calibrating for F-score; regressor calibration

All times in CEST. We thank **Meelis Kull** (U Tartu, Estonia) for his help in preparing this material.

References I

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

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

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

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, 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 II

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.

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.

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.

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 III

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.

Acknowledgements

- The work of MPN was supported by the SPHERE Next Steps Project funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/R005273/1.
- The work of PF and HS was supported by The Alan Turing Institute under EPSRC Grant EP/N510129/1.
- The work of MK was supported by the Estonian Research Council under grant PUT1458.
- The background used in the title slide has been modified by MPN from an original picture by Ed Webster with license CC BY 2.0.

Calibration: What and Why

Classifier Calibration Tutorial
ECML PKDD 2020

Prof. Peter Flach
Peter.Flach@bristol.ac.uk
classifier-calibration.github.io/

Departamento de ESTATÍSTICA

University of BRISTOL

UNIVERSITY OF TARTU