LLR calibration
what, why, how

Mauro Barni
Dept. Information Engineering and Mathematics
University of Siena
Siena, ITALY
PURDUE team

Goals

- Motivate the need for calibration
- Define a sound theoretical framework
- Overview baseline solutions
- Present open problems for Semafor
- Propose a roadmap for Semafor research

Working assumptions

- Hypothesis testing / two-class classifiers / binary detectors
  - H_0 = consistency check verified
  - H_1 = inconsistency found – something bad detected
  - Analytic outputs a score y such that:

$$\begin{array}{l} The\ larger\ y \\ the\ more\ evidence\ is\ found\ that \\ H_1\ holds\ (H_0\ is\ rejected) \end{array}$$

Notation

- Let f be an analytic, in the following I let:

$$x = input\ data$$
$$y = y_1 = f(x) = f_1(x)\ evidence\ in\ favor\ of\ H_1$$
$$t \in \{0, 1\}\ ground\ truth$$

- Often (e.g. with CNNs) f outputs two values

$$(y_0, y_1) \rightarrow y_0 = 1 - y_1 = 1 - y$$

- I also let: $P(H_0) = P_0, \quad P(H_1) = P_1$

What

- By calibration we refer to a procedure whereby the output of the analytic is given a

  precise probabilistic meaning

- Often we require that

$$Pr(H_1|f(x)=y)=y \quad \rightarrow Pr(H_0|f(x)=y)=1-y=y_0$$

- Other prob. quantities can be obtained from y

$$llr = log(y) - log(1-y) + log(P_0) - log(P_1)$$

Why calibration

- To make decisions based on error probabilities
- Minimum error probability obtained by

$$Reject\ H_0\ \ if\ y > 0.5$$

- Maximum likelihood decision

$$Reject\ H_0\ \ if\ \frac{y}{(1-y)}\frac{P_0}{P_1} > 1$$

- Compute P_f and P_m

Why calibration

- To let different analytics speak the same language
  - Ease fusion
- It can help to handle the variability of analytics
- It can help to cope with dataset (or domain) variability
  - By adapting the calibration dataset to the conditions at hand

Why calibration

- Express uncertainty
  - Distinguish between certain and uncertain decisions
  - Handle out of distribution data (e.g. by designing sound opt-out strategies)
  - Contrast DNN tendency to always output close-to-one values

On calibration datasets

- By its very definition, calibration requires the availability of either

  Probability models

  or

  Representative calibration datasets

- Calibration of $P(H_1|y)$ requires the availability of datasets generated both under H_0 and H_1

On calibration datasets

- Calibrating *llr* values also requires that datasets representative of both H_0 and H_1 are available
- The same applies to the calibration of both P_f and P_m
- Building representative datasets under H_1 may be difficult. In these cases calibration may be limited to P_f

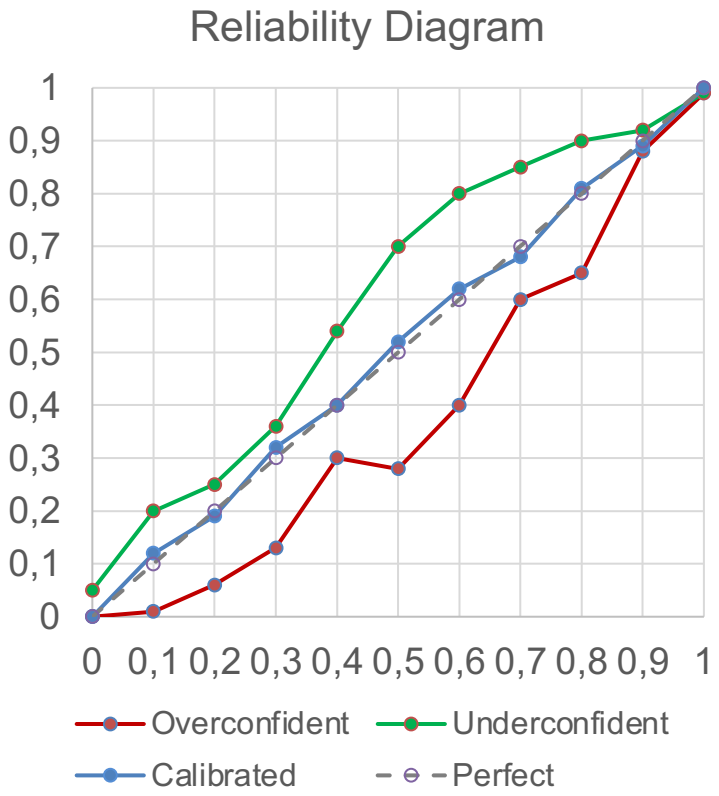
How

Assuming that a good calibration dataset is available covering both H_0 and H_1, several baseline approaches exist for calibration:

- Direct construction of calibrated analytics
  - Regularization during training
  - Bayesian networks
  - …
- *Post-hoc regularization*

Reliability diagrams

How should the output of a calibrated analytic look like?



In its simplest form, post-hoc calibration consists in applying a calibration function $g()$ to the network output y so that for $z = g(y)$ we have:

$$Pr(H_1|g(y) = z) = z$$

Parametric calibration

We assume a certain probability distribution (e.g. logistic) and optimize its parameter(s)

Simplest example: temperature scaling

Better seen on logits (ξ)

$$z = \frac{e^{\xi_1/T}}{e^{\xi_1/T} + e^{\xi_0/T}}$$

T is chosen so to maximize the likelihood of the observations under the pdf defined by z's

Platt scaling (logistic regression)

We assume the output probabilities are logistic functions of the scores

$$z = g(y) = \frac{1}{1 + e^{ay+b}}$$

Where a and b are determined by maximizing the likelihood or by fitting the logistic to the score obtained on the calibration set after binning

Platt scaling and *llr*

With logistic probabilities we have

$$llr = \log \frac{P(H1|z)}{P(H_0|z)} \frac{P(H_0)}{P(H_1)}$$

$$= \log \frac{1 + e^{(ay+b)}}{1 + e^{-(ay+b)}} + \log P(H_0) - \log P(H_1)$$

which is a shifted linear function in y hence allowing direct linear regression on llr

Note: the a-priori probabilities here correspond to the relative frequencies of the samples of the two classes in the calibration dataset

Other parametric calibration

Many other possibilities exist, for instance:

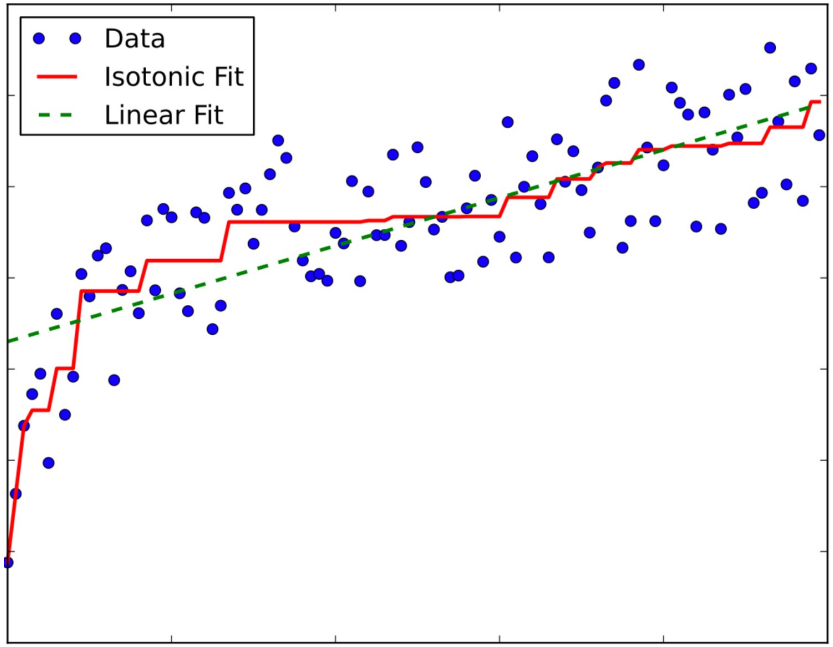
- **Vector and matrix scaling:** Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017 (vector and matrix scaling)
- **Histogram binning:** Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *NIPS*, 2019
- **Beta calibration**: Meelis Kull, Telmo M Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 2017.
- **CCAC (confidence calibration with auxiliary class)**: Shao, Z., Yang, J., & Ren, S. (2020). Calibrating Deep Neural Network Classifiers on Out-of-Distribution Datasets. arXiv preprint arXiv:2006.08914

Non-parametric calibration

Isotonic *regression is the baseline for non parametric calibration (assuming calibration set is large enough)*

After binning a piecewise linear non-decreasing function is fit to the calibration data

When population is increasing isotonic regression connects nearby points, otherwise it takes a constant value

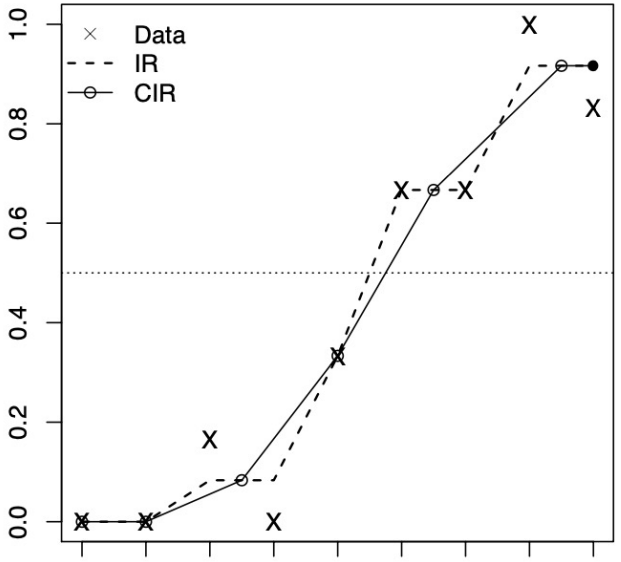


Centered isotonic regression

Flat regions typical of isotonic regression are not desirable

- Often probabilities (and confidence) should be strictly increasing
- Inversion of calibrated outputs is impossible, making it difficult to set a decision threshold

With centred isotonic calibration confidence values are strictly increasing



Oron, A. P., & Flournoy, N. (2017). Centered isotonic regression: point and interval estimation for dose–response studies. *Statistics in Biopharmaceutical Research*, *9*(3), 258-267.

Evaluating calibration

Expected Calibration Error

$$ECE = \sum_{i=1}^{n_{bin}} \frac{|B_i|}{N} \left| \hat{z}_{B_i} - \frac{|t=1|_i}{|B_i|} \right|$$

$$|t=1|_i = number\ of\ samples\ in\ B_i\ for\ which\ t=1$$

Maximum Calibration Error

$$MCE = \max_{i=1 \ldots n_{bin}} \left| \hat{z}_{B_i} - \frac{|t=1|_i}{|B_i|} \right|$$

POLITECNICO MILANO 1863

UNICAMP

Evaluating calibration (proper metrics)

Brier score

$$Brier\ score = \frac{1}{N}\sum_{i=1}^{N}(z_i - t_i)^2$$

Log-loss (cross-entropy, KL)

$$\frac{1}{N}\sum_{i=1}^{N} t_i \log z_i + (1 - t_i)\log(1 - z_i)$$

Operational metric

Difference between the performance achieved by setting the decision threshold based on the calibrated analytic and the best achievable performance obtained by setting the threshold on test data

From probabilities to *llr*

Passing from a calibrated output to a calibrated llr is easy

Passing to llr permits to remove the dependency on prior probabilities

$$llr = \log \frac{z}{1-z} + \log \frac{P(H_0)}{P(H_1)}$$

Where $P(H_0)$ and $P(H_1)$ are estimated based on the relative frequencies of the samples of the two classes in the calibration dataset

One-class calibration

If building a representative (calibration) dataset under H_1 is not possible we cannot calibrate llr and $P(H_{0/1}|y)$, *however we can still calibrate the probabilities under* H_0

Threshold calibration: choose T *in such a way that*

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(y_i \geq T) = P_f$$

Likelihood calibration: choose $z = g(y)$ *in such a way that*

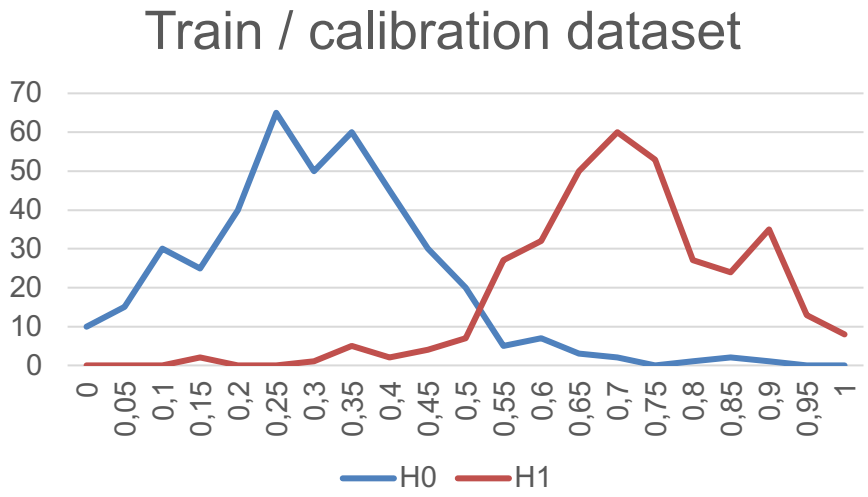
$$Pr(g(y) \leq z|H_0) = z$$

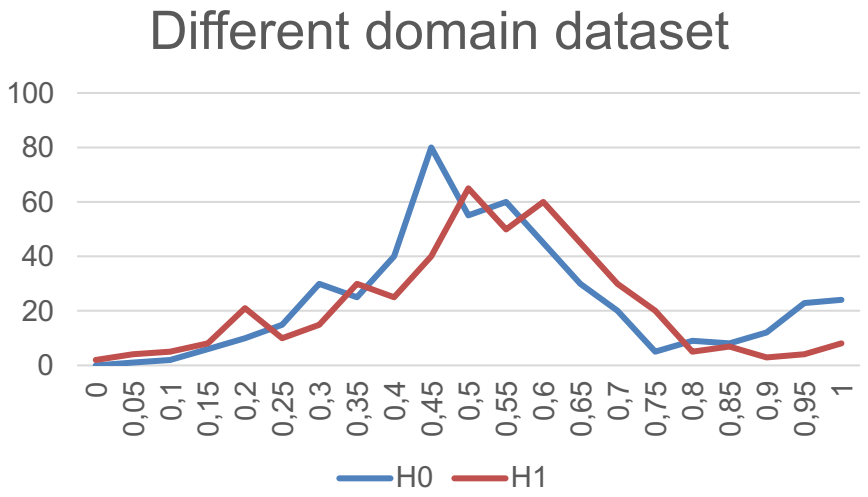
The BIGG problem

When the operative conditions can not be represented by one single dataset, single calibration procedures do not work

Two cases are possible

New analytic needed

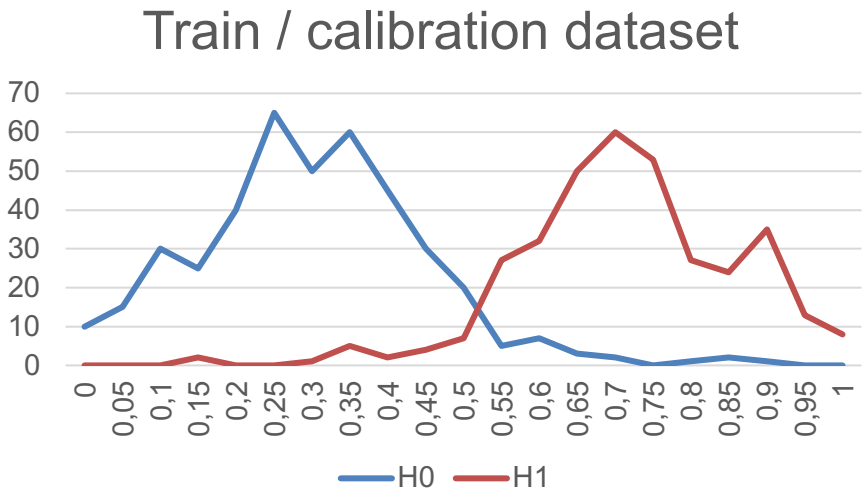


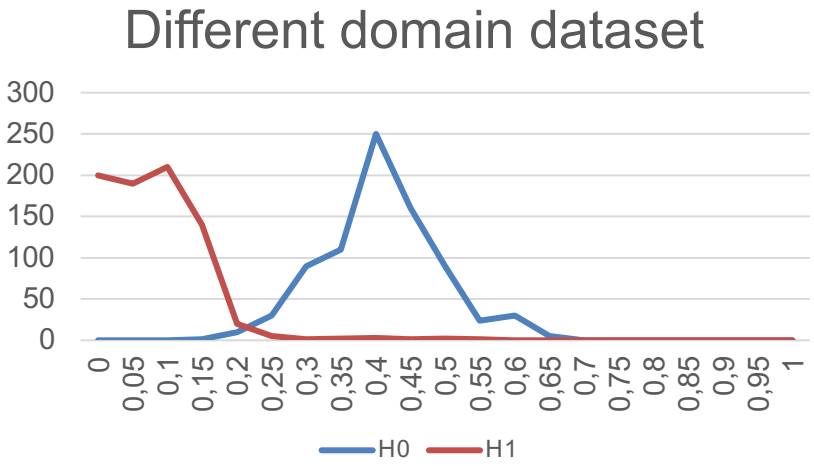


The features learnt during training are not effective in the new domain

Retraining (or fine tuning) needed

New calibration needed





The features learnt during training are still discriminative (AUC close to 1)

Recalibration is needed

Several possibilities

Define several domains/applications and calibrate (or train, if needed) a classifier for each domain

1. Identify domain and select calibration based on context information and/or metadata
2. Identify domain and select calibration based on the characteristics of input sample
3. Train e metaclassifier to identify domain
4. Include a rejection (opt-out option)

Roadmap

- Identify domains of interest (for different threats landscapes)
- Build calibration datasets
  - It is desirable that common calibration datasets are built
- Define single-dataset calibration procedures
  - Baselines + ad-hoc methods
- Define calibration metrics
  - Baseline + ad-hoc metrics
- Develop domain adaptive calibration procedures

Hope this presentation will help triggering further discussion and guide work ahead