# Model calibration

Alexander Lyzhov

# Why calibration?

# Overview

Classifier calibration definitions

Histogram regression estimators

Unbiased calibration estimation

Empirical results

Bonus: calibration methods

# How to define calibration?

$$\mathbb{P}\left[Y = \arg\max_y g_y(X) \mid \max_y g_y(X)\right] = \max_y g_y(X)^1$$ conf

$$\mathbb{P}[Y = y \mid g_y(X)] = g_y(X)$$ marginal

$$\mathbb{P}[Y = y \mid g(X)] = g_y(X)$$ joint

conf $\not\Rightarrow$ marginal

conf $+$ marginal $\not\Rightarrow$ joint

[1]Nixon, Jeremy, et al. "Measuring calibration in deep learning." arXiv preprint arXiv:1904.01685 (2019).

$$\mathbb{P}\left[Y = \arg\max_y g_y(X) \mid \max_y g_y(X)\right] = \max_y g_y(X)$$ conf

$$\mathbb{P}[Y = y \mid g_y(X)] = g_y(X)$$ marginal

conf $\not\Rightarrow$ marginal

| $g(X)$ | $\mathbb{P}[Y \in \cdot \mid g(X)]$ |
|---|---|
| $(\underline{0.6}, 0.1, 0.3)$ | $(\underline{0.7}, 0.2, 0.1)$ |
| $(0.4, \underline{0.6}, 0.0)$ | $(0.3, \underline{0.5}, 0.2)$ |

$$\mathbb{P}\left[Y = \arg\max_y g_y(X) \mid \max_y g_y(X)\right] = \max_y g_y(X)$$ conf

$$\mathbb{P}[Y = y \mid g_y(X)] = g_y(X)$$ marginal

$$\mathbb{P}[Y = y \mid g(X)] = g_y(X)$$ joint

conf + marginal $\not\Rightarrow$ joint

| $g(X)$ | $\mathbb{P}[Y \in \cdot \mid g(X)]$ |
|---|---|
| $(\underline{0.1}, 0.3, 0.6)$ | $(\underline{0.2}, 0.2, 0.6)$ |
| $(\underline{0.1}, 0.6, 0.3)$ | $(\underline{0.0}, 0.7, 0.3)$ |
| $(0.3, 0.1, 0.6)$ | $(0.2, 0.2, 0.6)$ |
| $(0.3, 0.6, 0.1)$ | $(0.4, 0.5, 0.1)$ |
| $(0.6, 0.1, 0.3)$ | $(0.7, 0.0, 0.3)$ |
| $(0.6, 0.3, 0.1)$ | $(0.5, 0.4, 0.1)$ |

$$\mathbb{P}[Y = y \mid g(X)] = g_y(X)$$ joint

joint $\not\Longrightarrow$ high accuracy

| $g(X)$ | $\mathbb{P}[Y \in \cdot \mid g(X)]$ |
|---|---|
| $(0.5, 0.5)$ | $(0.5, 0.5)$ |

| | $g(x)$ | $y$ |
|---|---|---|
| $x_1$ | $(0.5, 0.5)$ | $(1, 0)$ |
| $x_2$ | $(0.5, 0.5)$ | $(0, 1)$ |

# Calibration errors

$r(\xi) := (\mathbb{P}[Y = 1 \mid g(X) = \xi], \ldots, \mathbb{P}[Y = m \mid g(X) = \xi])$
Joint calibration $\iff$ $\mathrm{r}(\xi) = \xi$

$\mathrm{ECE} = \mathbb{E}[d(r(g(X)), g(X))]$
$\mathrm{MCE} = \sup [d(r(g(X)), g(X))]$
Joint calibration $\iff$ ECE=0, MCE=0

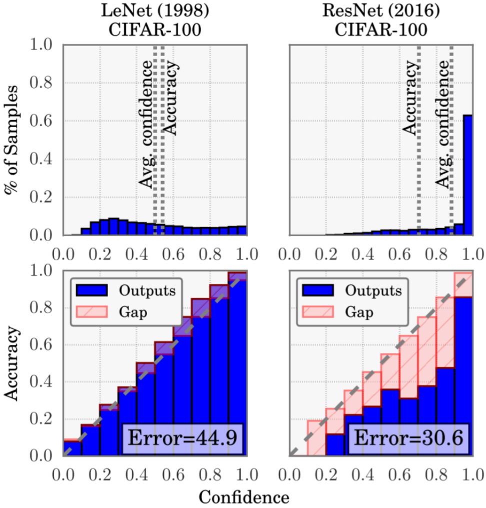
$r(\xi)$ is unknown. *How to estimate?*

# Histogram regression

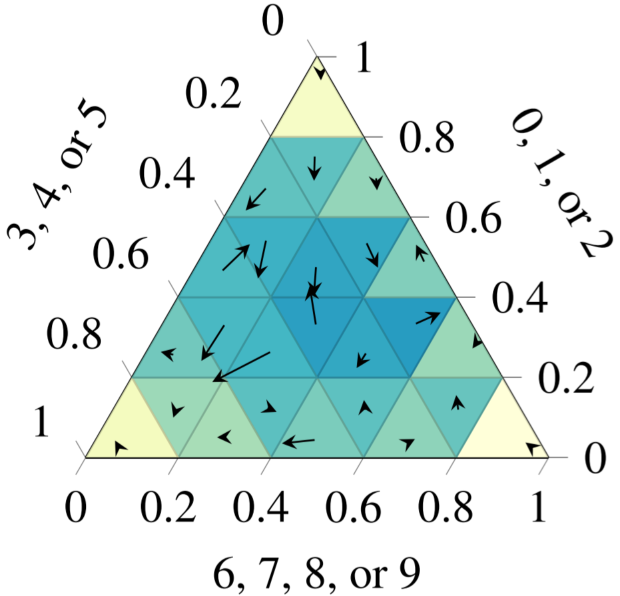
Partition simplex and let $b[\xi]$ be a partition corresponding to $\xi$.

Approximate $r_y(\xi) = \mathbb{P}(Y = y \mid g(X) = \xi)$ with:
$\hat{r}_y(\xi) := |\{i : g(x_i) \in b[\xi] \wedge y_i = y\}| \,/\, |\{i : g(x_i) \in b[\xi]\}|$

Approximate $\mathrm{ECE} = \mathbb{E}\left[d(r(g(X)), g(X))\right]$ with:
$\widehat{\mathrm{ECE}} = \sum_b \frac{n_b}{N} d\left(\frac{1}{n_b}\sum_b \hat{r}_y(g(x)), \frac{1}{n_b}\sum_b g(x)\right)$




[1]Guo, Chuan, et al. "On calibration of modern neural networks." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.




[1] Nixon, Jeremy, et al. "Measuring calibration in deep learning." arXiv preprint arXiv:1904.01685 (2019).

# Histogram regression pathologies

$\widehat{\mathrm{ECE}} = \sum_b \frac{n_b}{N} d\left(\frac{1}{n_b}\sum_b \hat{r}_y(g(x)), \frac{1}{n_b}\sum_b g(x)\right)$

$\lim_{\max_b \sup_{p,q\in b} \|p-q\|_2 \to 0} \lim_{n\to\infty} \widehat{\mathrm{ECE}} = \mathrm{ECE}$

$\lim_{n\to\infty} \widehat{\mathrm{ECE}} \leq \mathrm{ECE}$[1]

1. Biases for different models can be different
2. $\widehat{\mathrm{ECE}}$ is random, distribution is not taken into account
3. No common unit or scale
4. Depends a lot on binning scheme
5. Number of bins scales exponentially with number of classes

[1] Nixon, Jeremy, et al. "Measuring calibration in deep learning." arXiv preprint arXiv:1904.01685 (2019).

# Histogram regression pathologies

Example of $\widehat{\mathrm{ECE}}$ bias [1]:

$$\mathbb{P}(Y = 1) = \mathbb{P}(Y = 2) = 1/2$$
$$\mathbb{P}(Y = 0 \mid X = x) \in \{0, 1\} \; \forall x$$
$$g_{opt}(x) = \mathbb{P}(Y), g_{const}(x) \equiv (1/2, 1/2)$$

$$\mathrm{ECE}\left(g_{opt}\right) = 0, \mathrm{ECE}\left(g_{const}\right) = 0$$

One bin, single data point.

$$\mathbb{E}(\widehat{\mathrm{ECE}(\mathrm{g}_{\mathrm{opt}}})) = 0, \mathbb{E}(\widehat{\mathrm{ECE}(\mathrm{g}_{\mathrm{const}}})) = 1/2$$

[1] Nixon, Jeremy, et al. "Measuring calibration in deep learning." arXiv preprint arXiv:1904.01685 (2019).

# Unbiased calibration estimation

Instead of $\mathrm{ECE} = \mathbb{E}\left[d(r(g(X)), g(X))\right]$, define error as:
$\mathrm{CE}[\mathcal{F}, g] := \sup_{f \in \mathcal{F}} \mathbb{E}\left[(r(g(X)) - g(X))^\top f(g(X))\right]$[1]

joint $\implies \mathrm{CE} = 0$

$\mathrm{CE} = 0 \;\not\Longrightarrow$ joint

$\mathrm{CE} = 0, \mathcal{F} = C(\Delta^m \to \mathbb{R}^m) \implies$ joint

[1]Widmann, David, Fredrik Lindsten, and Dave Zachariah. "Calibration tests in multi-class classification: A unifying framework." arXiv preprint arXiv:1910.11385 (2019).

# Unbiased calibration estimation

RKHS $\mathcal{H} \iff$ kernel

Kernel $k : X \times X \to \mathbb{R}$,
$k(s,t) = k(t,s), \sum_{i,j=1}^{n} u_i u_j k(t_j, t_j) \geq 0$
Hilbert space $\{f : X \to \mathbb{R}\}$ is RKHS if $E_t f = f(t)$ is continuous.
Reproducing property: $f(t) = \langle f, k(\cdot, t) \rangle_{\mathcal{H}}$

Matrix-valued kernel $k : \Delta^m \times \Delta^m \to \mathbb{R}^{m \times m}$,
$k(s,t) = k(t,s)^T, \sum_{i,j=1}^{n} u_i^\top k(t_i, t_j) u_j \geq 0$
Hilbert space $\{f : \Delta^m \to \mathbb{R}^m\}$ is RKHS if $E_{t,u} f = \langle u, f(t) \rangle_{\mathbb{R}^m}$ is continuous.
Reproducing property: $\langle u, f(t) \rangle_{\mathbb{R}^m} = \langle k(\cdot, t) u, f \rangle_{\mathcal{H}}$

# Unbiased calibration estimation

Let $\mathcal{F}$ be unit ball in RKHS of $k$.
Kernel calibration error:
$\mathrm{KCE}[k, g] := \mathrm{CE}[\mathcal{F}, g] =$
$\sup_{f \in \mathcal{F}} \mathbb{E}\left[(r(g(X)) - g(X))^\top f(g(X))\right]$

$\mathrm{KCE} = 0 \iff$ joint

$$\mathrm{KCE} = \left(\mathbb{E}\left[(e_Y - g(X))^\top k\left(g(X), g\left(X'\right)\right)\left(e_{Y'} - g\left(X'\right)\right)\right]\right)^{\frac{1}{2}}$$

No $r(g(X))$ now!

# Unbiased calibration estimation

We had $\mathrm{CE}[\mathcal{F}, g] := \sup_{f \in \mathcal{F}} \mathbb{E}\left[(r(g(X)) - g(X))^\top f(g(X))\right]$
$\mathrm{KCE}[k, g] := \mathrm{CE}[\mathcal{F}, g]$

$$\mathrm{KCE} = \left(\mathbb{E}\left[\left(e_Y - g(X)\right)^\top k\left(g(X), g\left(X'\right)\right)\left(e_{Y'} - g\left(X'\right)\right)\right]\right)^{\frac{1}{2}}$$

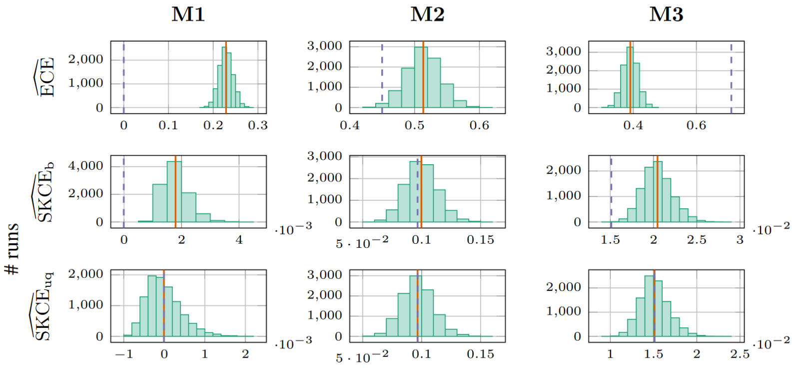
$h_{i,j} := \left(e_{Y_i} - g\left(X_i\right)\right)^\top k\left(g\left(X_i\right), g\left(X_j\right)\right)\left(e_{Y_j} - g\left(X_j\right)\right)$

| Notation | Definition | Properties | Complexity |
|---|---|---|---|
| $\widehat{\mathsf{SKCE}}_{\mathrm{b}}$ | $n^{-2} \sum_{i,j=1}^{n} h_{i,j}$ | biased | $O\left(n^2\right)$ |
| $\widehat{\mathsf{SKCE}}_{\mathrm{uq}}$ | $\binom{n}{2}^{-1} \sum_{1 \leq i < j \leq n} h_{i,j}$ | unbiased | $O\left(n^2\right)$ |
| $\widehat{\mathrm{SKCE}}_{\mathrm{ul}}$ | $\lfloor n/2 \rfloor^{-1} \sum_{i=1}^{\lfloor n/2 \rfloor} h_{2i-1,2i}$ | unbiased | $O(n)$ |

Potentially trainable.

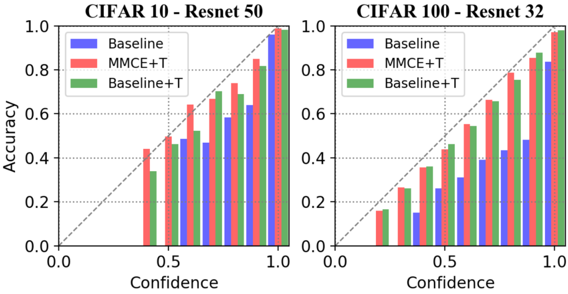
# Results

$$k(x, y) = \exp(-\|x - y\|/\nu)\mathbf{I}$$



# Results

$$\mathrm{conf}_i = \max g(y_i \mid x_i)$$
$$\mathrm{correct}_i = \mathbb{I}\left[\hat{y}_i = \arg\max g(y_i \mid x_i)\right]$$
$$\mathrm{SKCE}_\theta(B) = \sum_{i,j \in B} \frac{(\mathrm{conf}_i - \mathrm{correct}_i)\left(\mathrm{conf}_j - \mathrm{correct}_j\right) k\left(r_i, r_j\right)}{|B|^2}$$ [1]
$$\min_\theta \sum_{(x_i,y_i)\in B} \log g_\theta\left(y_i \mid x_i\right) + \lambda \left(\mathrm{SKCE}_\theta\left(B\right)\right)^{\frac{1}{2}}$$




[1] Kumar, Aviral, Sunita Sarawagi, and Ujjwal Jain. "Trainable calibration measures for neural networks from kernel mean embeddings." International Conference on Machine Learning. 2018.

# Bonus: calibration methods

1. Non-parametric methods

Histogram binning[1]:
$\min_{\theta_1,\ldots,\theta_M} \sum_{m=1}^{M} \sum_{i=1}^{n} \mathbf{1}\left(a_m \leq \hat{p}_i < a_{m+1}\right)\left(\theta_m - y_i\right)^2$

Isotonic regression[2]:
$\min_{\theta,a} \sum_{m=1}^{M} \sum_{i=1}^{n} 1\left(a_m \leq \hat{p}_i < a_{m+1}\right)\left(\theta_m - y_i\right)^2$
$0{=}\mathrm{a}_1 \leq a_2 \leq \ldots \leq a_{M+1} = 1, \theta_1 \leq \theta_2 \leq \ldots \leq \theta_M$

---

[1] Zadrozny, Bianca, and Charles Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers." Icml. Vol. 1. 2001.

[2] Zadrozny, Bianca, and Charles Elkan. "Transforming classifier scores into accurate multiclass probability estimates." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 2002.

# Bonus: calibration methods

2. Parametric methods

Platt scaling: Softmax($Wz + b$) [1]

Temperature scaling: Softmax($z/T$)

+ of temperature scaling:
Easy and very effective for conf
Doesn't affect accuracy

- of temperature scaling:
Requires validation set
Changes confidence uniformly
Doesn't work well for marginal

[1] Platt, John. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." Advances in large margin classifiers 10.3 (1999): 61-74.

# Bonus: calibration methods

3. Calibration during training

Optimize kernel calibration error

Regularize entropy of predictive distribution[1]

Use focal loss instead of NLL: FL(p) = $-(1-g_{y_i}(x))^{\gamma} \log g_{y_i}(x)$[2]

[1] Pereyra, Gabriel, et al. "Regularizing neural networks by penalizing confident output distributions." arXiv preprint arXiv:1701.06548 (2017).

[2] Mukhoti, Jishnu, et al. "Calibrating Deep Neural Networks using Focal Loss." arXiv preprint arXiv:2002.09437 (2020).

# Open questions

- How to optimize estimates of marginal/joint calibration while training?
- How to pick the kernel? Can we learn it?
- Does this help in safety-critical applications?
- Do existing recalibration techniques for NNs result in better marginal/joint calibration?
- How do NN hyperparameters influence kernel calibration error?