16 Jan 2020

Q Why is probability calibration required ?

Probabilities predicted by some models are not well calibrated. They can be over confident in some cases or under confident in some cases.

In case of data imbalance, model may over favor the majority class.

In general, it is good idea to calibrate predicted probabilities for non linear ml models prior to evaluating their performance.

Q Which algorithm return well calibrated probabilities ?

Logistic regression return well calibrated probabilities when classes are balanced. When classes are not balanced, then even LR require calibration.

Q What does calibrated probability mean?

Calibrated probability means that the probability reflects the likelihood of true events.

Predicted probabilities = if expected distribution of probabilities of each class

Calibrated probabilities.

Q which models produce uncalibrated predicted probabilities?

→ Non linear ml models
→ Imbalanced data
→ LR in case of imbalanced data.

Q Why complex models do not produce well calibrated probabilities?

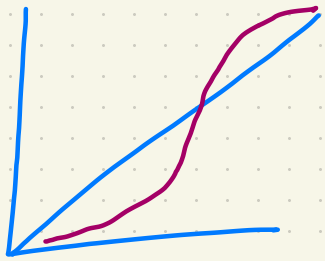
Because they use approximations instead of probabilistic predictions.

Q When is calibration process done?

It is a rescaling operation that is done after predicted probabilities are generated by the model.

Q How do you interpret reliability curves?

Straight line denotes ideal behavior.



For points below the line, model is overestimating.

For points above the line, model is underestimating.

Q What do reliability diagrams do?

It provides a diagnostic tool to check whether the forecast value is reliable or not.

Q What are different ways of probability calibration?

→ Platt scaling

→ Isotonic regression

Platt Scaling -

→ Simple

→ Use when reliability curve is S-shape / sigmoid shaped.

Isotonic Regression

→ Complex, require more data otherwise overfit when data is scarce.

→ supports any shape

Q Does calibration of probabilities always lead to better results?

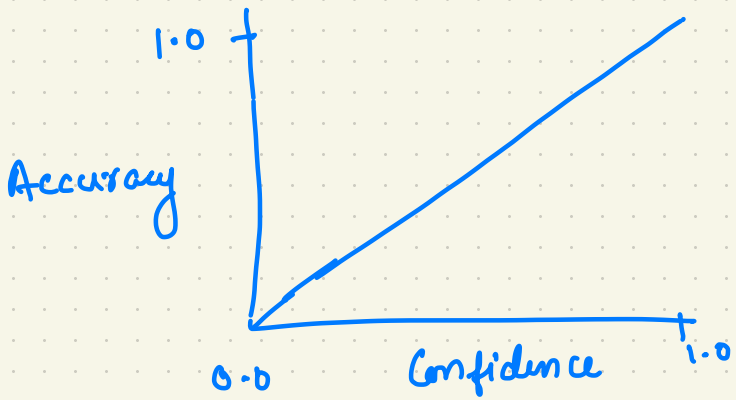
It may or may not. SVMs, bagged DTs, RF benefit from calibration.

Q When is model said to be perfectly calibrated?

$$P(\hat{y} = 1 \mid \hat{p} = p) = p, \quad \forall\ p \in [0,1]$$

a model is said to be perfectly calibr-

- ated if and only if, for any $p \in [0,1]$ prediction of a class with confidence p is correct 100 percent of the time.



Q Define accuracy & confidence in various bins ?

Let's group the predictions into M bins, each of size of $1/M$. Let B_m be the set of indices of samples whose predicti on confidence falls into interval I_m.

$$I_m = \left(\frac{m-1}{M}, \frac{m}{M}\right], \text{ for } m \in \{1, \ldots, M\}$$

accuracy of B_m :

$$acc(B_m) = \frac{1}{|B_m|} \sum 1(\hat{y}_i = y_i)$$

Average confidence in B_m $\quad conf_{B_m} = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$

where $\hat{p}_i$ is confidence of sample i.

Q. Expected Calibration Error (ECE)?

It is computed as weighted average of accuracy / confidence difference.

$$ECE = \sum_{m=1}^{M} \frac{|B_M|}{N} |acc(B_m) - conf(B_m)|$$

Q Maximum calibration error?

Empirically it is computed as :-

$$MCE = \max_{m \in \{1, \ldots M\}} |acc(B_m) - conf(B_m)|$$

Q why calibration is important in DNN?

→ modern DNN focus on reducing cross entropy loss to improve classification accuracy. It is observed that these models obtain better classification accuracy at expense of well modeled probabilities.

→ In production, DNNs are used as a decision making component.

→ Because minimizing a cross entropy loss does not ensure calibration, and even tends to overfit classification accuracy, it is imperative to calibrate any model where probabilities are passed on to other decision making system.

Q. Name some calibration methods?

→ Platt scaling

→ Temperature scaling

Each method takes form of an additional model that corrects calibration error of an original model.

Fitting the calibration model is a distinct step, done only after original model is trained.

Both approaches make use of validation set for purpose of fitting a calibration model.

Platt scaling :-

→ applies to binary classification

$$\hat{q}_i = \sigma(a z_i + b) \quad \text{where}$$

z_i is logit for example i (the original network's pre activation output)

a, b are scalar parameters optimized using a cross entropy loss over validation set.

Temperature scaling

→ Works for multiclass classification.

$$\hat{q}_i = \max_k \sigma_{SM}(z_i^k / T)$$

σ_{SM} = softmax function

T = tunable temperature parameter