Calibration for probabilistic classification

Nick

Overview

The problem

some classifiers produce well-calibrated probabilities

- discriminant analysis
- logistic regression

others don't

- naive bayes
- SVMs
- anything with boosting
- tree methods
- sometimes neural networks

First of all, who cares?

1. people with asymmetric misclassification costs
2. people who are going to use the scores in post-processing
3. people who want to compare model outputs on a fair basis

Definitions: “classification”

in general, a classifier is a mapping function f such that

$$f : \vec{x} \mapsto c$$

where $\vec{x} \in \mathbb{R}^P$, but we're mostly interested in the intermediate step in where the function produces some membership score s_i for each instance $\vec{x}_i$

Definitions: "well-calibrated"

- for a model f and score s_i to be well-calibrated for class c_i, the empirical probability of a correct classification $P(c_i|f(c_i|x_i) = s_i)$ must converge to $f(c_i|x_i) = s_i$
- **example**: when $s_i = 0.9$, the probability of a correct classification should converge to $P(c_i|s_i = 0.9) = 0.9$. Otherwise, this isn't *really* a 'probability.'

Definitions: "calibration"

the calibration process is a separate mapping such that

$$g : s_i \mapsto P(c_i|s_i)$$

it's really important to note that we're fitting another model on top of our model output, where your feature matrix is just the vector of probability scores $\vec{s}$ and the target variable is the vector of true class labels $\vec{y} \in \{0, 1\}$

Common methods

Platt scaling

Pass s_i through the sigmoid

$$P(c_i|s_i) = \frac{1}{1 + \exp(As_i + B)}$$

where A and B are the solution to

$$\operatorname*{argmax}_{A,B} - \sum_i y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

Isotonic regression

A strictly-nondecreasing piecewise linear function m, where

$$y_i = m(s_i) + \epsilon$$

fit such that

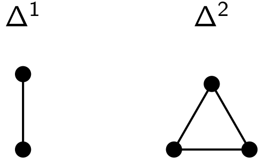
$$\hat{m} = argmin_z \sum_i y_i - z(s_i)^2$$

Extensions to $k > 2$

Probabilistic classification as a simplex

- if we view the task of probabilistic classification as a vector-valued function, we can visualize the co-domain of this task as assigning the location of a prediction in a regular (unit) simplex, Δ^{K-1}
- why is this hard when $K > 2$?

Probabilistic classification as a simplex



trivial with Δ^1 because we're only concerned with one unknown value and its complement. With $\Delta^{K>2}$ the simplex becomes a triangle, tetrahedron, five-cell, etc.

Multi-class probability estimation

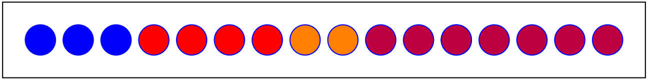


Figure 1: classification problem with $k = 4$

Strategy: decompose into separate binary classification problems

- one vs. all
- all pairs

One vs. all

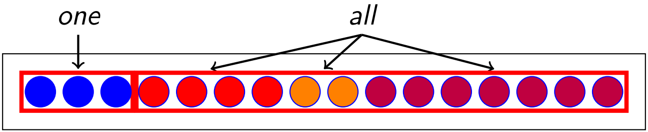


Figure 2: *one vs. all* reduces to $k - 1$ calibrations

All pairs

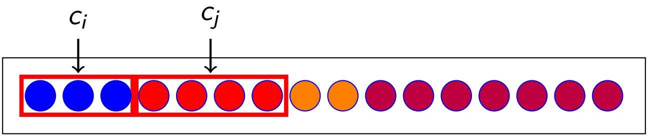


Figure 3: *all pairs* reduces to $\binom{K}{2}$ calibrations

Combining multi-class probability estimates

Experimental results

Conclusion

References