accuracy

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

Accuracy is the fraction of correct predictions made by a hypothesis on a dataset with a finite label space, equal to one minus the average 0/1 loss. It is widely used as a single-number metric for classification methods. Accuracy is not suitable as an objective function for training, since the 0/1 loss is non-smooth and non-convex. Empirical risk minimization (ERM) uses a smooth surrogate loss instead, and accuracy is reported during validation. On imbalanced data, a large accuracy can hide failure on the minority class, which calls for other means of evaluation such as the confusion matrix.

Definition

Accuracy is the fraction of correct predictions made by a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ on a dataset $\mathcal{D} = \big\{\big(\mathbf{x}^{(r)}, y^{(r)}\big)\big\}_{r=1}^{m}$ with a finite label space $\mathcal{Y}$ (Goodfellow et al., 2016, Sect. 5.1):

$$\begin{aligned}\mathrm{acc}(h|\mathcal{D}) &:= 1 - \frac{1}{m}\sum_{r=1}^{m} L^{(0/1)}\Big(\big(\mathbf{x}^{(r)}, y^{(r)}\big), h\Big) \\ &= \frac{1}{m}\Big|\Big\{r \in \{1, \ldots, m\} : h\big(\mathbf{x}^{(r)}\big) = y^{(r)}\Big\}\Big|.\end{aligned}$$

Accuracy ranges from 0 (no prediction is correct) to 1 (every prediction is correct). Equivalently, accuracy equals one minus the average 0/1 loss on the dataset. For example, in image classification, accuracy is the fraction of images assigned the correct category.

Fig. 1 compares two classifiers on the same dataset. The linear classifier h misclassifies three data points near its decision boundary. The nonlinear classifier h' curves around each of these data points and classifies them correctly.

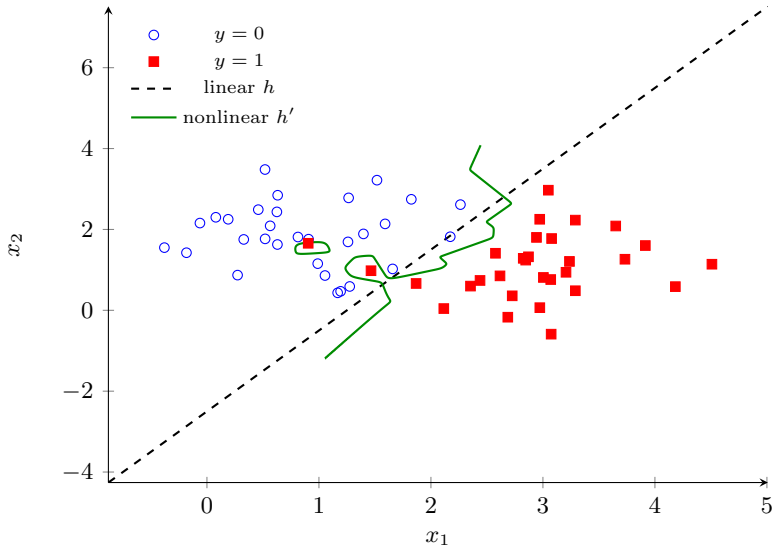

Figure 1: Two classifiers applied to the same $m = 60$ data points with classes $y \in \{0, 1\}$. The linear classifier h (dashed) places three data points on the wrong side of its decision boundary, yielding $\text{acc}(h|\mathcal{D}) = 57/60 = 0.95$. The nonlinear classifier h' (solid green) curves around those three data points and yields $\text{acc}(h'|\mathcal{D}) = 1$. Data generated by `pythondemos/accuracy.py`

The classifier h' in Fig. 1 achieves the optimal accuracy of 1 on $\mathcal{D}$: its decision boundary closely follows the positions of the individual data points that the linear classifier misclassifies. As a result, h' is sensitive to small perturbations of the feature vectors, such as measurement noise. A slight shift in the feature vector of a data point near the decision boundary can flip the prediction. Optimizing accuracy on the training set therefore does not by itself ensure good generalization.

While accuracy is often used to compare learned classifiers on a validation set or a test set, it is not suitable as an objective function for training. Indeed, accuracy is defined via the 0/1 loss, which is non-smooth and non-convex. Gradient-based optimization methods cannot maximize accuracy directly. Instead, training methods for classifiers, e.g., via empirical risk minimization (ERM), typically use a surrogate loss that approximates the 0/1 loss. Two widely used examples for such a surrogate loss are the logistic loss, which is convex and differentiable, and the hinge loss, which is convex but not differentiable. Fig. 2 shows the three loss functions for binary classification with $\mathcal{Y} = \{-1, 1\}$, where a real-valued hypothesis h predicts via the sign of $h(\mathbf{x})$. Each loss function depends on the data point only through the product $y \cdot h(\mathbf{x})$.

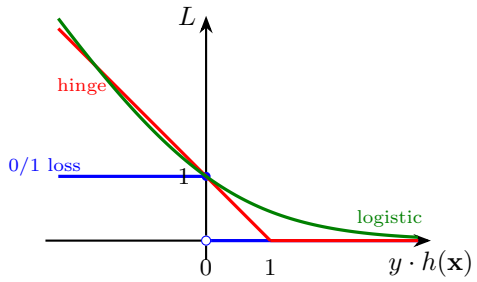

Figure 2: Three loss functions for binary classification as functions of the product $y \cdot h(\mathbf{x})$. The hinge loss and logistic loss are convex upper bounds on the 0/1 loss

Accuracy can also be misleading for imbalanced data. Consider a frost warning for Krems an der Donau (Austria): the features of a data point are the minimum and the maximum air temperature of one day, and its label indicates whether the minimum temperature of the following day falls below 0 °C. Frost days are the minority class: only 6 of 40 recorded days are followed by frost. The constant prediction "above zero" (a baseline that ignores the features) is therefore correct on 34 of the 40 days, an accuracy of 0.85, and a classifier learned by logistic regression from the 40 data points reaches the same accuracy of 0.85. Accuracy cannot tell the two apart, which calls for other means of evaluation such as the confusion matrix. The confusion matrices in Fig. 3 show that the learned classifier detects one of the six frost days while the baseline detects none.

learned classifier: accuracy 0.85

true \ predicted	above 0	frost
above 0	33	1
frost	5	1

always-above-zero baseline: accuracy 0.85

true \ predicted	above 0	frost
above 0	34	0
frost	6	0

Figure 3: The confusion matrices of the frost-warning classifier learned by logistic regression and of the always-above-zero baseline. Both reach an accuracy of 0.85; only the confusion matrices show that the learned classifier detects one frost day and raises one false alarm, while the baseline detects none. Data generated by `pythondemos/cm.py`

Cross References

0/1 loss, loss, classification, F_1 score, confusion matrix, imbalanced data

References

1. Goodfellow et al. (2016). Deep Learning. MIT Press, Cambridge, MA, USA.