logistic regression

Authors

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

Abstract

Logistic regression learns a linear hypothesis map $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ for a binary classification problem with label $y \in \{-1, 1\}$. The sign of $h(\mathbf{x})$ is the predicted label, and the sigmoid function transform $\sigma(h(\mathbf{x}))$ estimates the probability of the label $y = 1$. The parameters are learned by empirical risk minimization (ERM) with the average logistic loss on the training set, which is equivalent to maximum likelihood estimation under the sigmoid function probability model. The resulting objective is convex and smooth, so gradient descent (GD) converges to a minimizer whenever one exists; on a linearly separable training set a penalty term restores a unique minimizer. Thresholding the learned hypothesis yields a linear classifier whose decision boundary is a hyperplane.

Definition

An email service must decide, for every incoming message, whether to move it to the spam folder. Features extracted from the message — word frequencies, sender information — form its feature vector $\mathbf{x} \in \mathbb{R}^d$, and the label $y \in \{-1, 1\}$ records whether the message is spam. Logistic regression learns, for such a binary classification problem, a linear hypothesis map $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ (Bishop, 2006; Hastie et al., 2009). Despite its name, the method solves a classification problem — the regression in the name refers to fitting the real-valued map h, not to a numeric label.

The value $h(\mathbf{x})$ is read in two ways. Its sign is the predicted label, which makes the learned h a linear classifier whose decision boundary is the hyperplane $\mathbf{w}^\top \mathbf{x} = 0$. Its sigmoid function transform $\sigma(h(\mathbf{x}))$ is an estimate of the probability of the label $y = 1$ given the feature vector. The quality of a candidate $\mathbf{w}$ is measured by the average logistic loss on the training set $\mathcal{D}^{(\text{train})} = \{(\mathbf{x}^{(r)}, y^{(r)})\}_{r=1}^m$ of m data points, and empirical risk minimization (ERM) delivers

$$\widehat{\mathbf{w}} \in \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}), \text{ with } f(\mathbf{w}) := \frac{1}{m} \sum_{r=1}^{m} \log \Big(1 + \exp\big(-y^{(r)} \mathbf{w}^\top \mathbf{x}^{(r)}\big)\Big).$$

Minimizing f is equivalent to maximum likelihood estimation under the model that assigns the label $y = 1$ with probability $\sigma(\mathbf{w}^\top \mathbf{x})$ (Bishop, 2006). Fig. 1 shows a learned probability curve for a training set with a single feature.

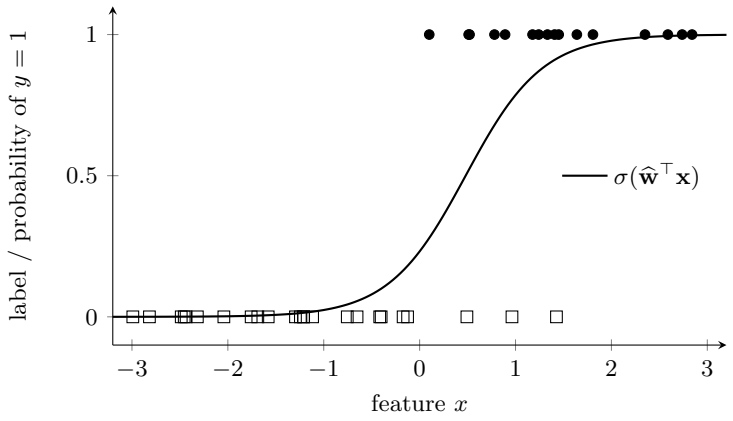

Figure 1: Logistic regression on a training set with a single feature. Data points with label $y = 1$ are drawn as filled circles at height one, those with $y = -1$ as open squares at height zero. The curve is the sigmoid function transform $\sigma(\widehat{\mathbf{w}}^\top \mathbf{x})$ of the learned hypothesis: it estimates the probability of the label $y = 1$ and crosses $1/2$ at the decision boundary. Data generated by `pythondemos/logreg.py`

Unlike linear regression with the squared error loss, the minimization admits no closed-form solution, but f is convex and smooth, so gradient descent (GD) applies. The iteration is

$$\mathbf{w}^{(t+1)} = T\big(\mathbf{w}^{(t)}\big), \text{ with } T(\mathbf{w}) := \mathbf{w} - \eta \nabla f(\mathbf{w}),$$

with learning rate η and the gradient $\nabla f(\mathbf{w})$. A vector $\mathbf{w}$ is a fixed point of the operator T exactly when $\nabla f(\mathbf{w}) = \mathbf{0}$, which by convexity of f is exactly when $\mathbf{w}$ is a minimizer. For a sufficiently small learning rate, no update increases f, and the iterates converge to a minimizer whenever one exists (Hastie et al., 2009). On a linearly separable training set no minimizer exists — the norm of the iterates grows without bound while the training error tends to zero — and adding a penalty term to f, one of the three routes that regularization distinguishes, restores a unique minimizer.

Applying linear regression with the squared error loss to the binary labels is a near miss: the squared error loss penalizes a prediction $\mathbf{w}^\top \mathbf{x}$ that is large, positive, and correct, while the logistic loss decreases as this margin grows. For more than two label values, replacing the sigmoid function by the softmax function yields multinomial logistic regression (Bishop, 2006).

Cross References

classification, binary classification, linear classifier, decision boundary, sigmoid function, logistic loss, ERM, GD, maximum likelihood, linear regression, softmax function, regularization

References

1. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

2. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.