

linear classifier

Author and editors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland

Editors: Konstantina Oliumtsevs, Ekkehard Schnoor, Aalto University, Espoo, Finland

*Corresponding author: alex.jung@aalto.fi

Abstract

A linear classifier is a classifier whose decision regions are separated by hyperplanes. For binary classification it predicts the sign of a linear model hypothesis $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, whose decision boundary is the hyperplane with normal vector $\mathbf{w}$. The hypothesis value $h(\mathbf{x})$ is the distance of $\mathbf{x}$ from that hyperplane, multiplied by $\|\mathbf{w}\|_2$, and its sign names the side. That distance bounds the feature perturbations the prediction tolerates. The model parameters are learned by empirical risk minimization (ERM) with a convex surrogate for the 0/1 loss: logistic regression uses the logistic loss and the support vector machine (SVM) the hinge loss. More than two label values are handled by one linear model hypothesis per value and the largest-score rule, which makes every decision region an intersection of halfspaces.

Definition

Mapping the vineyards of a wine region starts from an aerial photograph. The photograph is cut into square patches of 25 m side length, and each patch becomes a data point. Two numbers describe a patch: its contrast x_1 , the standard deviation of the brightness of its pixels, and its greenness x_2 , the relative difference between the average green and the average red channel value, in percent. The two numbers form the feature vector $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$, and the label is $y = +1$ for a patch showing a vineyard and $y = -1$ otherwise. Vine rows alternate foliage with bare soil, so a vineyard patch is more contrasted and less green than the forest and meadow around it. A straight line separates most of the vineyard patches from the rest (Fig. 1).

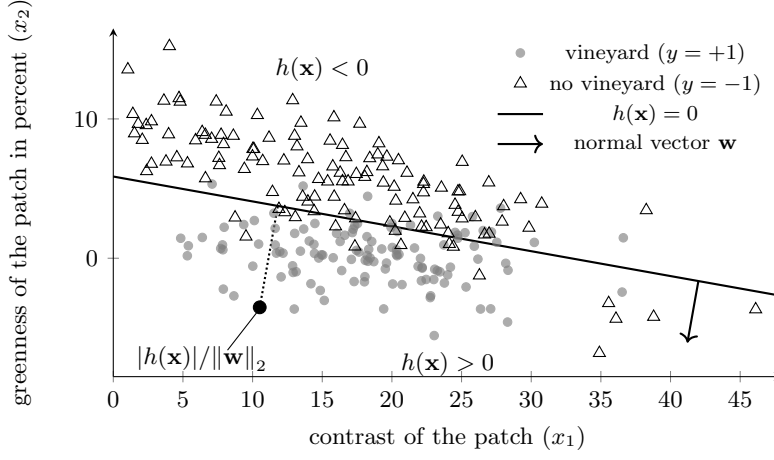


Figure 1: Data points of the vineyard map: each patch of the aerial photograph is placed at its contrast (x_1) and its greenness (x_2). The solid line is the decision boundary $h(\mathbf{x}) = 0$ of a linear classifier learned from these data points, the arrow is its normal vector $\mathbf{w}$, and the dotted segment drops one vineyard patch (filled black circle) perpendicularly onto the decision boundary. The length of that segment is $|h(\mathbf{x})|/\|\mathbf{w}\|_2$. The feature vectors are drawn from a two-component Gaussian model whose means and covariance matrices are the patch statistics of an aerial photograph of the Wachau valley. Data generated by `pythondemos/linclass.py`

A linear classifier is a classifier whose decision regions are separated by hyperplanes in $\mathbb{R}^d$ (Bishop, 2006, Sect. 4.1.1; Shalev-Shwartz and Ben-David, 2014, Sect. 9.1). For binary classification, with label space $\mathcal{Y} = \{-1, +1\}$, such a classifier is obtained from a linear model. A hypothesis

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b,$$

with model parameters $\mathbf{w} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and $b \in \mathbb{R}$, delivers the prediction $\hat{y} = \text{sign}(h(\mathbf{x}))$. Its decision boundary is the hyperplane

$$\{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{x} + b = 0\},$$

with normal vector $\mathbf{w}$, and the two decision regions are the two halfspaces it bounds.

The number $h(\mathbf{x})$ carries more than its sign. Consider a feature vector $\mathbf{x}$ and any vector $\mathbf{x}'$ of the decision boundary. Subtracting $\mathbf{w}^\top \mathbf{x}' + b = 0$ from $h(\mathbf{x})$ gives

$$h(\mathbf{x}) = \mathbf{w}^\top (\mathbf{x} - \mathbf{x}').$$

The Cauchy-Schwarz inequality bounds the right-hand side in magnitude by $\|\mathbf{w}\|_2 \|\mathbf{x} - \mathbf{x}'\|_2$. Every vector of the decision boundary is therefore at distance at

least $|h(\mathbf{x})|/\|\mathbf{w}\|_2$ from $\mathbf{x}$. That distance is attained by the orthogonal projection

$$\mathbf{x}' = \mathbf{x} - \frac{h(\mathbf{x})}{\|\mathbf{w}\|_2^2} \mathbf{w},$$

which satisfies $\mathbf{w}^\top \mathbf{x}' + b = 0$. The distance of $\mathbf{x}$ from the decision boundary is thus $|h(\mathbf{x})|/\|\mathbf{w}\|_2$, and the hypothesis value $h(\mathbf{x})$ is that distance multiplied by $\|\mathbf{w}\|_2$, with the sign naming the side (Bishop, 2006, Sect. 4.1.1; Hastie et al., 2009, Sect. 4.5, Eq. (4.40)). Fig. 1 shows this distance for one vineyard patch.

The hypothesis value by itself does not measure that distance. Let s be a positive number. Replacing the model parameters $(\mathbf{w}, b)$ by $(s\mathbf{w}, sb)$ leaves the decision boundary and every prediction unchanged, while every hypothesis value is multiplied by s . Dividing by $\|\mathbf{w}\|_2$ removes this freedom, which is why the distance and not $h(\mathbf{x})$ is the quantity with a geometric meaning.

The distance also bounds the feature perturbations that a prediction tolerates. Perturbing $\mathbf{x}$ by a vector $\boldsymbol{\delta}$ changes the hypothesis value by $\mathbf{w}^\top \boldsymbol{\delta}$, which the Cauchy-Schwarz inequality bounds in magnitude by $\|\mathbf{w}\|_2 \|\boldsymbol{\delta}\|_2$. The sign of $h(\mathbf{x})$, and with it the prediction, therefore survives every perturbation with $\|\boldsymbol{\delta}\|_2 < |h(\mathbf{x})|/\|\mathbf{w}\|_2$. A vineyard patch far from the decision boundary keeps its prediction under a measurement error that flips the prediction for a patch close to it. Multiplying $h(\mathbf{x})$ by the label y gives the margin, which is positive exactly on the correctly classified data points. For a linearly separable training set and a sufficiently small regularization parameter α , the support vector machine (SVM) learns the linear classifier whose smallest margin is largest (Shalev-Shwartz and Ben-David, 2014, Sect. 15.1, Eq. (15.1)).

The model parameters $(\mathbf{w}, b)$ are learned from a training set $\mathcal{D}^{(\text{train})} = \{(\mathbf{x}^{(r)}, y^{(r)})\}_{r=1}^m$ by empirical risk minimization (ERM). Counting the mistakes with the 0/1 loss gives an objective function that is piecewise constant, and minimizing it is computationally hard outside the linearly separable case (Shalev-Shwartz and Ben-David, 2014, Sect. 9.1). Practical methods replace the 0/1 loss by a convex loss of the margin: logistic regression uses the logistic loss and the SVM uses the hinge loss. Both learn a hypothesis from the same hypothesis space, the linear model, and differ only in the loss. Both objective functions are convex: gradient descent (GD) minimizes the smooth one of logistic regression, and subgradient descent with a subgradient the non-smooth one of the SVM. Adding a penalty term $\alpha \|\mathbf{w}\|_2^2$ to the average loss, one of the three routes that regularization distinguishes, keeps $\|\mathbf{w}\|_2$ small and the margins wide (Shalev-Shwartz and Ben-David, 2014, Sect. 15.2).

For more than two label values, with $\mathcal{Y} = \{1, \dots, K\}$, a linear classifier uses one linear model hypothesis $g_c(\mathbf{x}) = (\mathbf{w}^{(c)})^\top \mathbf{x} + b^{(c)}$ per label value $c \in \mathcal{Y}$ and predicts the value with the largest score, $\hat{y} = \arg \max_{c \in \mathcal{Y}} g_c(\mathbf{x})$. The decision region $\mathcal{R}_c$ is then an intersection of $K - 1$ halfspaces, one per competing label value, and is therefore a convex set; the border between $\mathcal{R}_c$ and $\mathcal{R}_{c'}$ lies in the hyperplane on which the two scores agree (Bishop, 2006, Sect. 4.1.2, Eq. (4.10)). The resulting partition of the feature space is shown in Fig. 2.

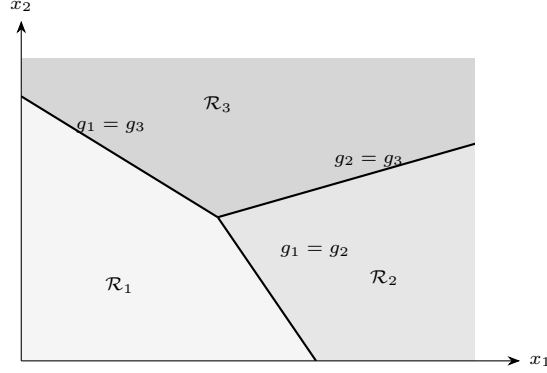


Figure 2: A linear classifier for $K = 3$ label values partitions the feature space $\mathbb{R}^2$ into three decision regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$. Each border lies in the hyperplane on which the two competing scores agree, and each decision region is an intersection of halfspaces and therefore convex

A linear classifier is linear in the features, not in the raw measurements a data point comes with. Applying a feature transformation $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ first and a linear classifier to $\phi(\mathbf{x})$ afterwards gives a decision boundary that is a hyperplane in $\mathbb{R}^{d'}$ and a curved surface in the original feature space. A kernel method constructs such a feature transformation implicitly, and a deep net learns one from the training set, its last layer being a linear classifier on the features the earlier layers compute (Bishop and Bishop, 2024, Sect. 6.3.3).

Cross References

classifier, binary classification, decision boundary, decision region, hyperplane, halfspace, linear model, margin, logistic regression, SVM, ERM, 0/1 loss, feature transformation

References

1. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.
2. Bishop and Bishop (2024). Deep Learning: Foundations and Concepts. Springer Nature, Cham, Switzerland.
3. Shalev-Shwartz and Ben-David (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Univ. Press, New York, NY, USA.
4. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.