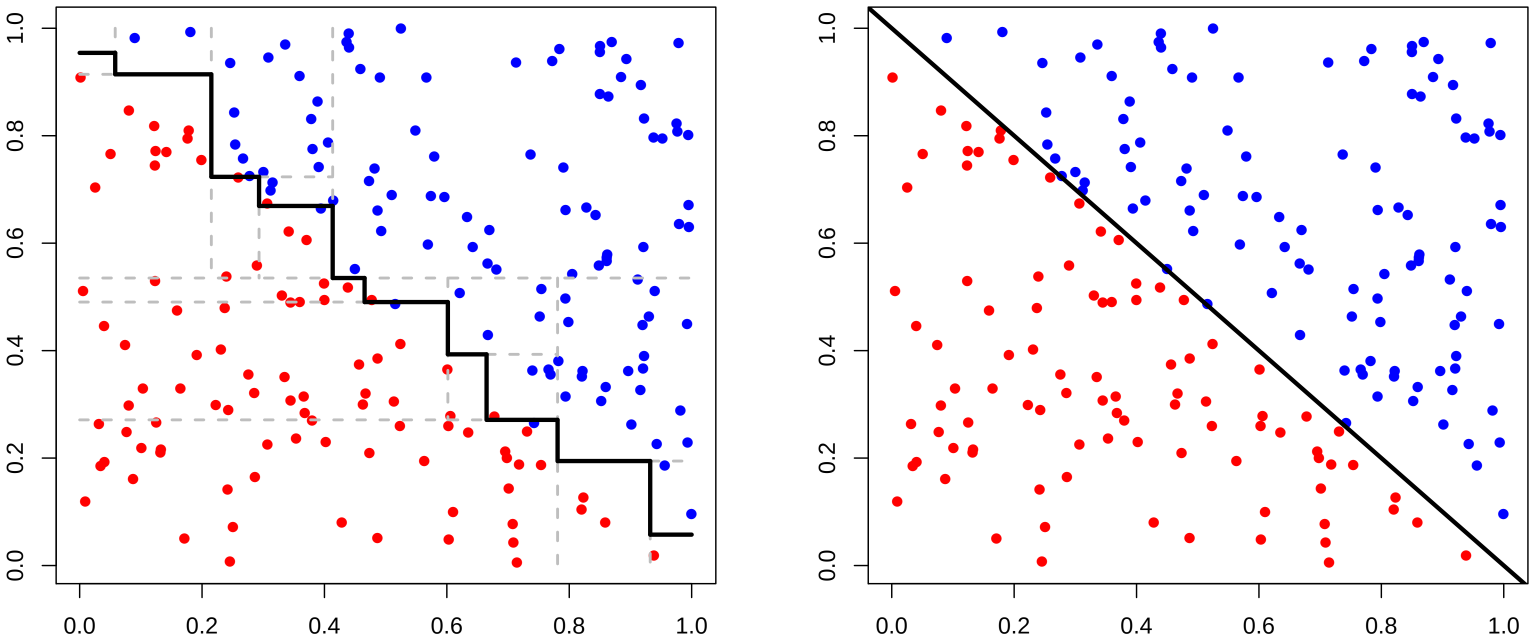
MTAT.03.227 Machine Learning

Linear classification

Sven Laur
University of Tartu

Why linear classification is useful?



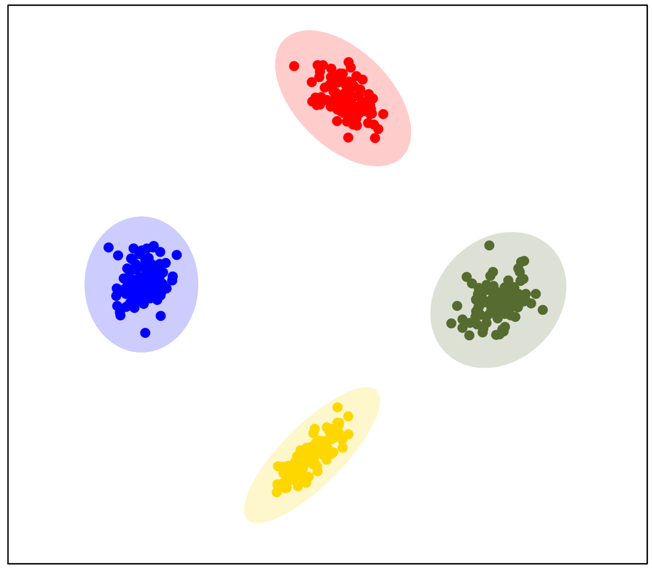
Decision trees split data into smaller chunks with thresholding $x_i \leq t_i$.

▷ Thus decision trees cannot track linear boundaries well.

▷ Linear decision borders $w_1 x_1 + \cdots + w_n x_n \leq -w_0$ occur often.

▷ Linear decision borders make chunking easier for recursive splitting.

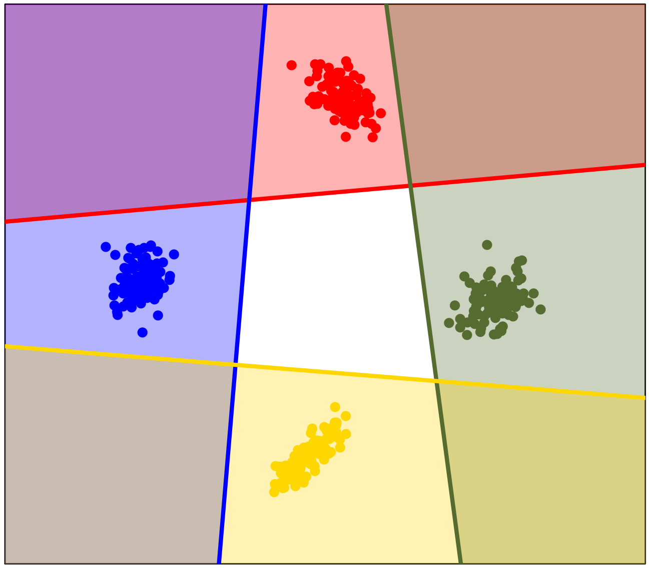
From binary to arbitrary classification



Binary classification can be used for multi-label classification problems:

- ▷ one-vs-all classification
- ▷ all-vs-all classification
- ▷ hierarchical classification

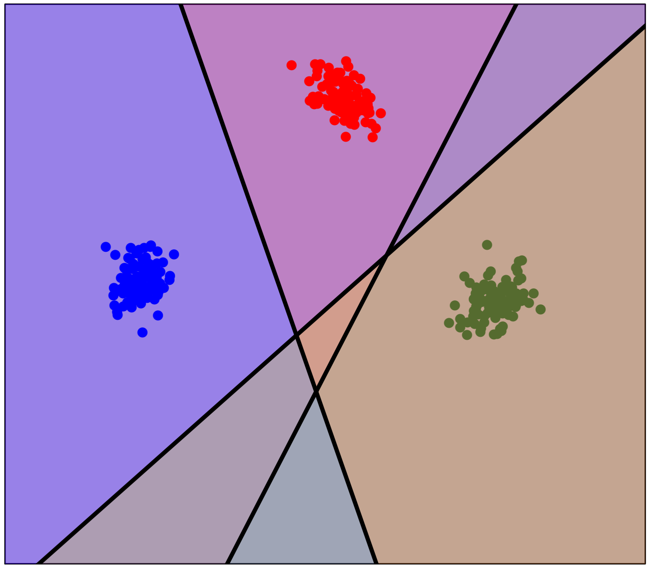
One versus all classification



Classification is obtained by a committee voting:

- ▷ A one-vs-all classifier for each potential label.
- ▷ The label with the highest decision value wins.

All versus all classification



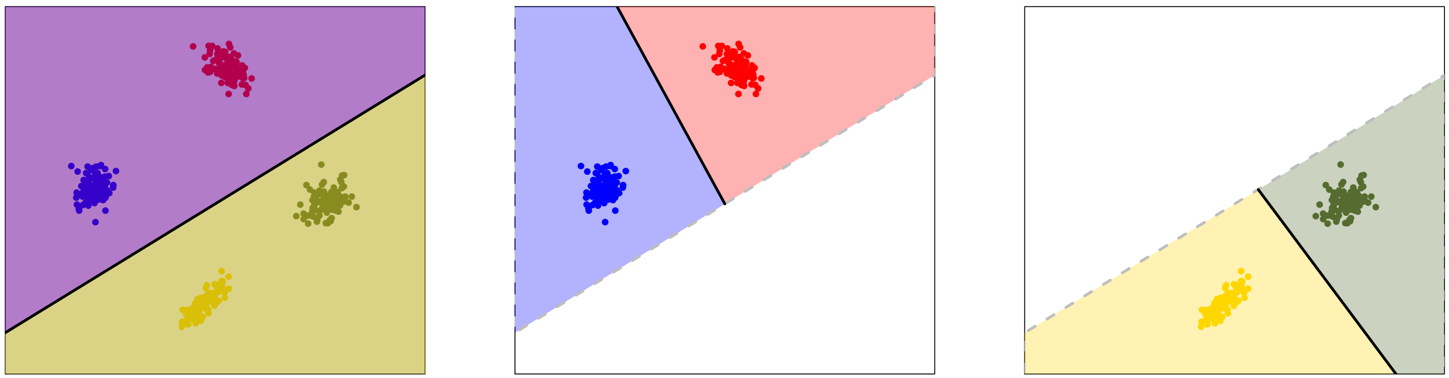
Classification is obtained by a committee voting:

▷ A one-vs-other classifier for each potential label pair.

▷ Votes for each label are aggregated.

▷ The label with the highest number of votes wins.

Hierarchical classification



Hierarchical classification resolves labels gradually:

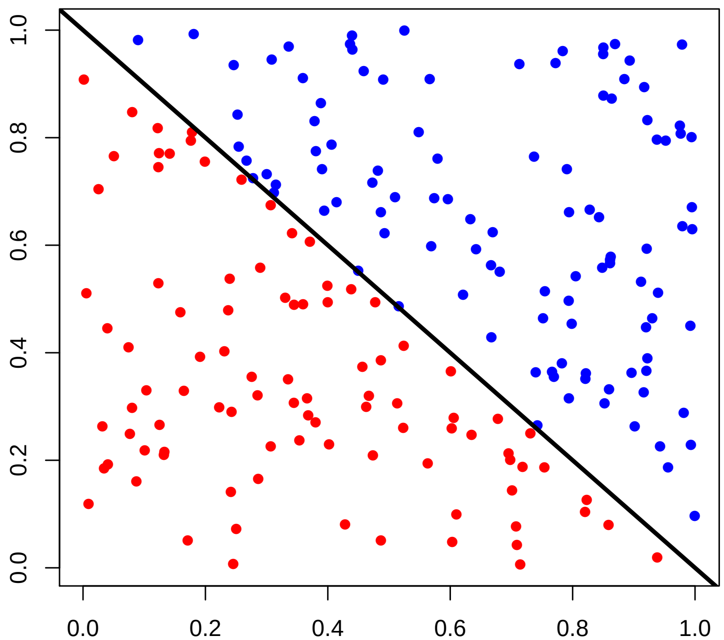
▷ First all labels are split into two groups.

▷ Then each group is recursively split until two labels remain.

▷ The final class is decided in a leaf node with a binary classifier.

Differently from decision trees a label can be produced only by a single leaf.

Binary linear classification



Decision rule

$$y = \mathrm{sign}(w_0 + w_1 x_1 + \cdots w_n x_n)$$

Discriminant function

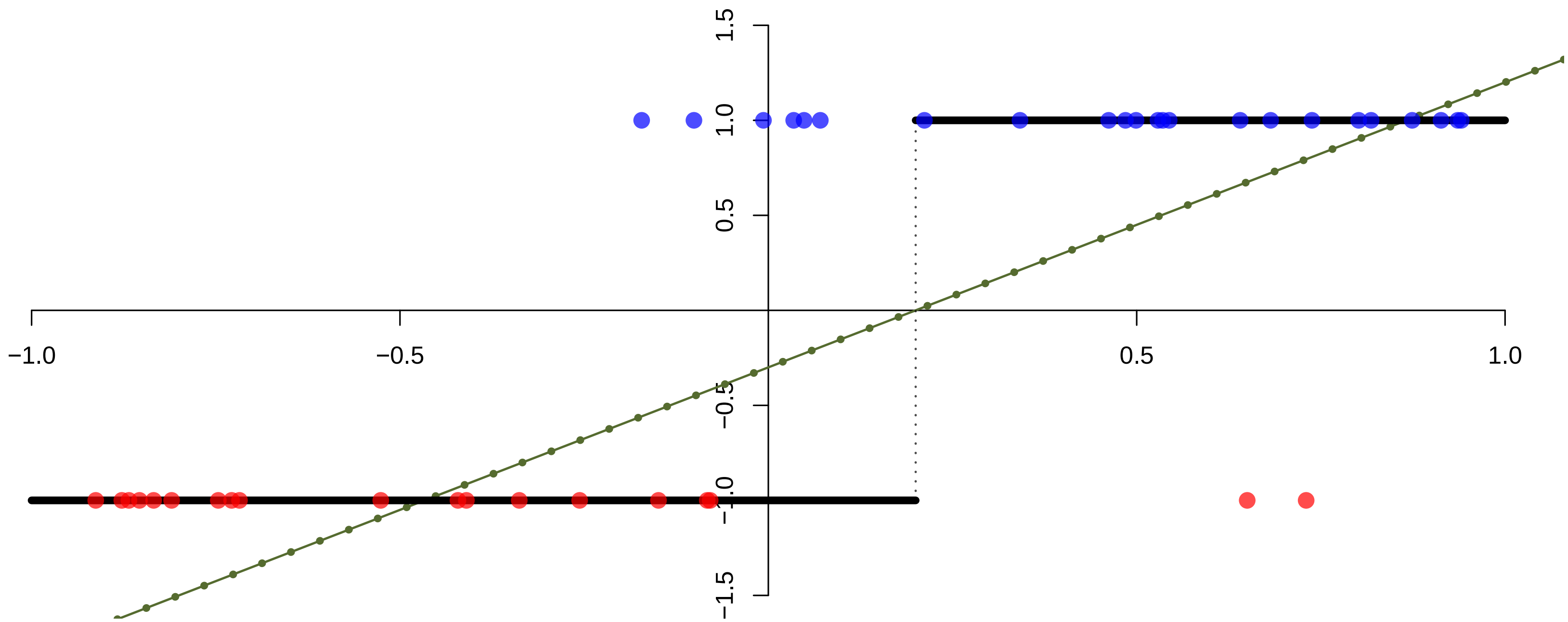
$$f(\boldsymbol{x}) = w_0 + w_1 x_1 + \cdots w_n x_n$$

Model parameters

- ▷ bias term w_0
- ▷ weights $\boldsymbol{w} = (w_1, \ldots, w_n)^{\mathrm{T}}$

- ▷ Model parameters are found by minimising a cost function.
- ▷ Different cost functions lead to a different linear classifiers.
- ▷ Different linear models have same limitations but different performance.

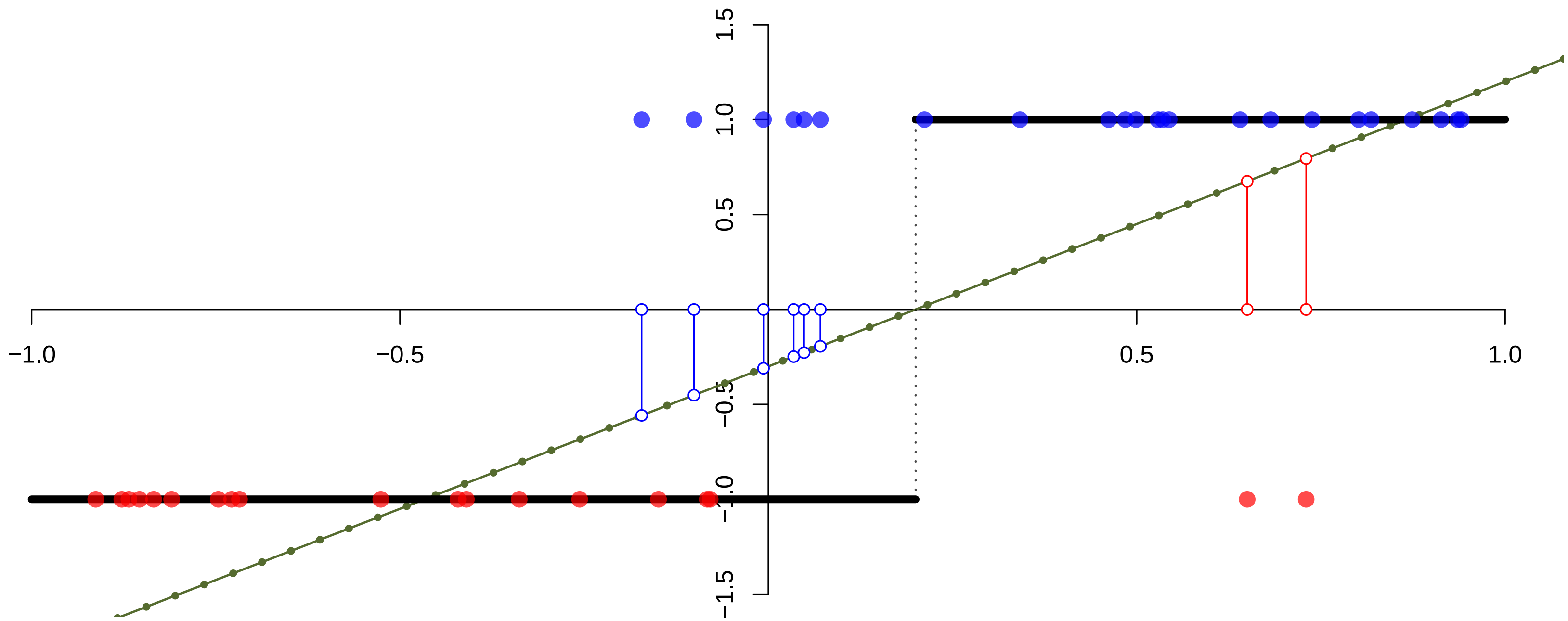
How to choose a cost function?



We cannot use the number misclassified labels as a cost function

- ▷ Cost function must be almost everywhere differentiable
- ▷ Cost function should have only one global minimum point

Perceptron cost function



Not all mistakes are equal. Mistakes with high values of $f_{\boldsymbol{w},w_0}(\boldsymbol{x}_i)$ are bad

$$J(\boldsymbol{w}, w_0) = \sum_{\{i: f_{\boldsymbol{w},w_0}(\boldsymbol{x}_i) y_i < 0\}} -y_i f_{\boldsymbol{w},w_0}(\boldsymbol{x}_i) = \sum_{\{i: f_{\boldsymbol{w},w_0}(\boldsymbol{x}_i) y_i < 0\}} -y_i(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x_i} + w_0)$$

Perceptron cost is continuous

The cost function is continuous and bounded from below by zero.

▷ For simplicity consider alternative formulation of the function

$$J(\boldsymbol{w}, w_0) = -\sum_{i=1}^{n}[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i} + w_0) < 0]y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i} + w_0)$$

and look at individual terms of the sum.

▷ If a decisions is certain: $|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + w_0| > 0$ then in a small neighbourhood $\|\delta\boldsymbol{w}\|^2 + \delta w_0^2 < \varepsilon$ the predicted labels $\hat{y}$ do not change and the continuity follows from the continuity of sum and scalar product.

▷ For a borderline decision $|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + w_0| = 0$, we can show that $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + w_0$ converges to zero if the neighbourhood shrinks $\varepsilon \to 0$. This is enough!

Gradient of the perceptron cost

The cost function is differentiable in all points where the decision is certain

$$\frac{\partial J}{\partial \boldsymbol{w}} = -\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{w}} [y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i} + w_0) < 0] y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i} + w_0)$$

$$= -\sum_{i=1}^{n} [y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i} + w_0) < 0] y_i \frac{\partial \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x_i}}{\partial \boldsymbol{w}}$$

$$= -\sum_{\{i: f_{\boldsymbol{w}, w_0}(\boldsymbol{x_i}) y_i < 0\}} y_i \boldsymbol{x}_i$$

$$\frac{\partial J}{\partial w_0} = -\sum_{\{i: f_{\boldsymbol{w}, w_0}(\boldsymbol{x_i}) y_i < 0\}} y_i$$

In borderline points we can use the formulae above to fix the gradient value.

Perceptron cost has no local minima

Let $\boldsymbol{w}, w_0$ be a local minimum. Then for any $\alpha \in \mathbb{R}$ the predicted labels of $f_{\boldsymbol{w},w_0}$ and $f_{\alpha\boldsymbol{w},\alpha w_0}$ coincide. Consequently,

$$J_{\alpha\boldsymbol{w},\alpha w_0} = \alpha J_{\boldsymbol{w},w_0}$$

and $\boldsymbol{w}, w_0$ can be a local minimum only if $J_{\boldsymbol{w},w_0} = 0$.

- ▷ Does the cost function have non-trivial minimum?
  - ◇ If labels are linearly separable such a minimum exists
  - ◇ Gradient decent will converge to this point for non-zero starting point.
- ▷ Does gradient decent converge to such a minimum in finite time?
  - ◇ Certain gradient decent algorithms converge in finite number of steps
  - ◇ Simple proofs exists for fixed learning rate

Minimum region for the perceptron cost

The global mimimum is achievable in a large region

Minimum squared error cost

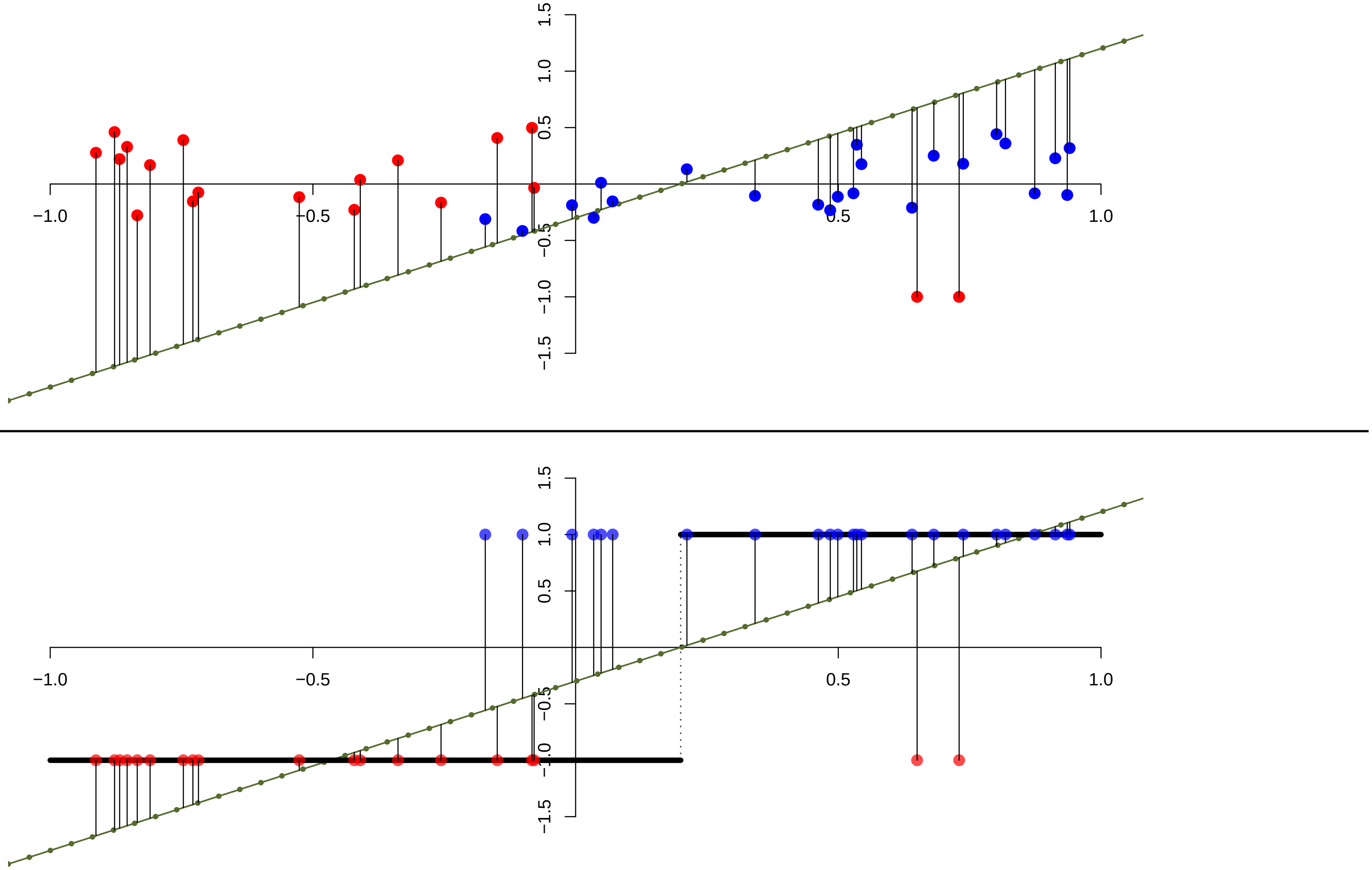
If we define the classification penalty

$$J_{\boldsymbol{w},w_0} = \sum_{i=1}^{n} (y_i^* - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - w_0)^2$$

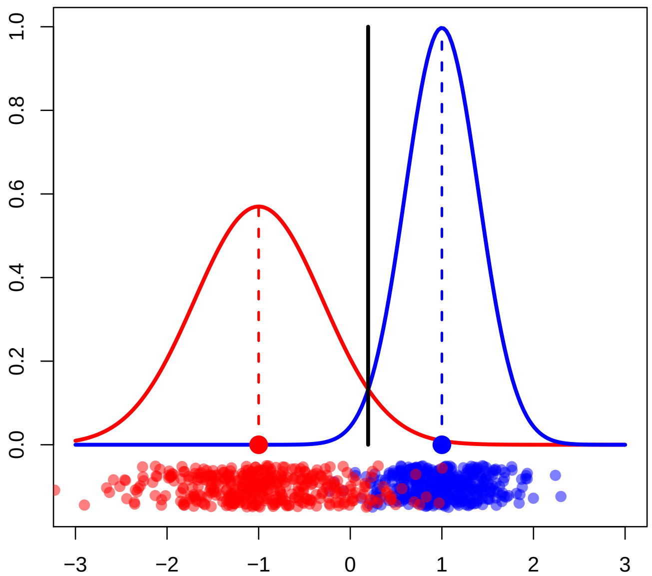
where y_i^* are computable from true labels y_i then

- ▷ The existence of a single minimum is guaranteed.
- ▷ The solution can be expressed by closed linear regression formula.
- ▷ However, the solution is not guaranteed to have a good precision.
- ▷ For certain values of $y_1^*, \ldots, y_n^*$ the solution is known to be sensible.

Linear regression vs linear classification



Fishers cost function for 1D



Mean values

$$m_j = \frac{1}{n_1} \sum_{y_i = j} x_i$$

Variance

$$\sigma_j^2 = \frac{1}{n_1} \sum_{y_i = j} (x_i - m_j)^2$$

- ▷ Estimate mean and standard deviation of both classes.
- ▷ Find the threshold so that both density functions are equal.