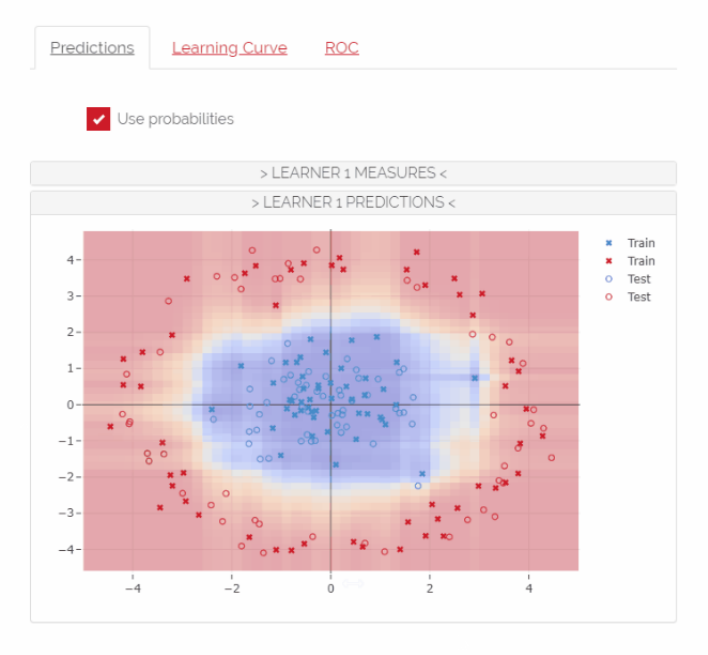
Introduction to Machine Learning

Gradient Boosting for Classification



Learning goals

- Transfering gradient boosting for regression to binary classification problems
- Introducing gradient boosting for multiclass problems

BINARY CLASSIFICATION

For $\mathcal{Y} = \{0, 1\}$, we simply have to select an appropriate loss function, so let us use Bernoulli loss as in logistic regression:

$$L(y, f(\mathbf{x})) = -y \cdot f(\mathbf{x}) + \log(1 + \exp(f(\mathbf{x}))).$$

Then,

$$\begin{aligned} \tilde{r}(f) &= -\frac{\partial L(y, f(\mathbf{x}))}{\partial f(\mathbf{x})} \\ &= y - \frac{\exp(f(\mathbf{x}))}{1 + \exp(f(\mathbf{x}))} \\ &= y - \frac{1}{1 + \exp(-f(\mathbf{x}))} = y - s(f(\mathbf{x})). \end{aligned}$$

Here, $s(f(\mathbf{x}))$ is the logistic function, applied to a scoring model. Hence, effectively, the pseudo-residuals are $y - \pi(\mathbf{x})$.
Through $\pi(\mathbf{x}) = s(f(\mathbf{x}))$ we can also estimate posterior probabilities.

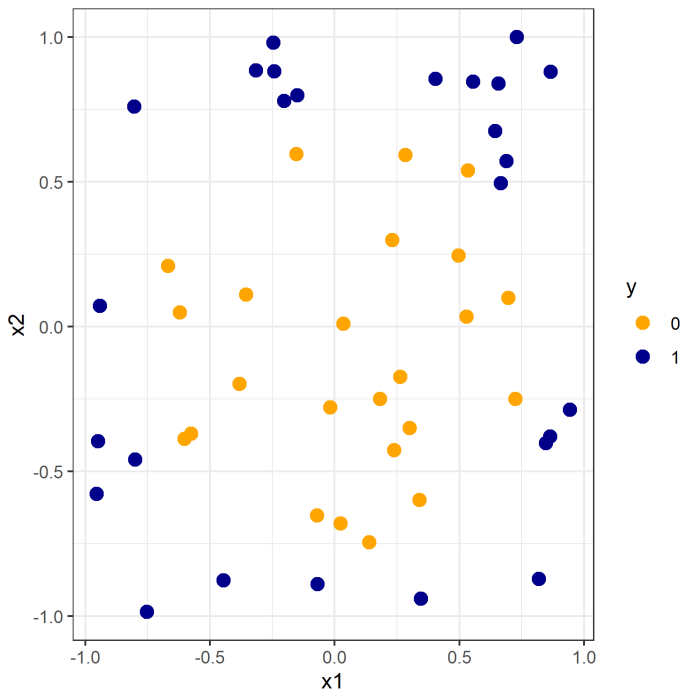
BINARY CLASSIFICATION

- Using this loss function, we can simply run GB as for regression. NB: Also here, we fit regression base learners against our numerical vector of pseudo-residuals with *L2* loss.
- We could also have used the exponential loss for classification with GB. It can be shown that the resulting GB algorithm is basically equivalent to AdaBoost. In practice there is no big difference, although Bernoulli loss makes a bit more sense from a theoretical (maximum likelihood) perspective.
- It follows that GB is a generalization of AdaBoost which can also use other loss functions and be used for different ML scenarios.

EXAMPLE

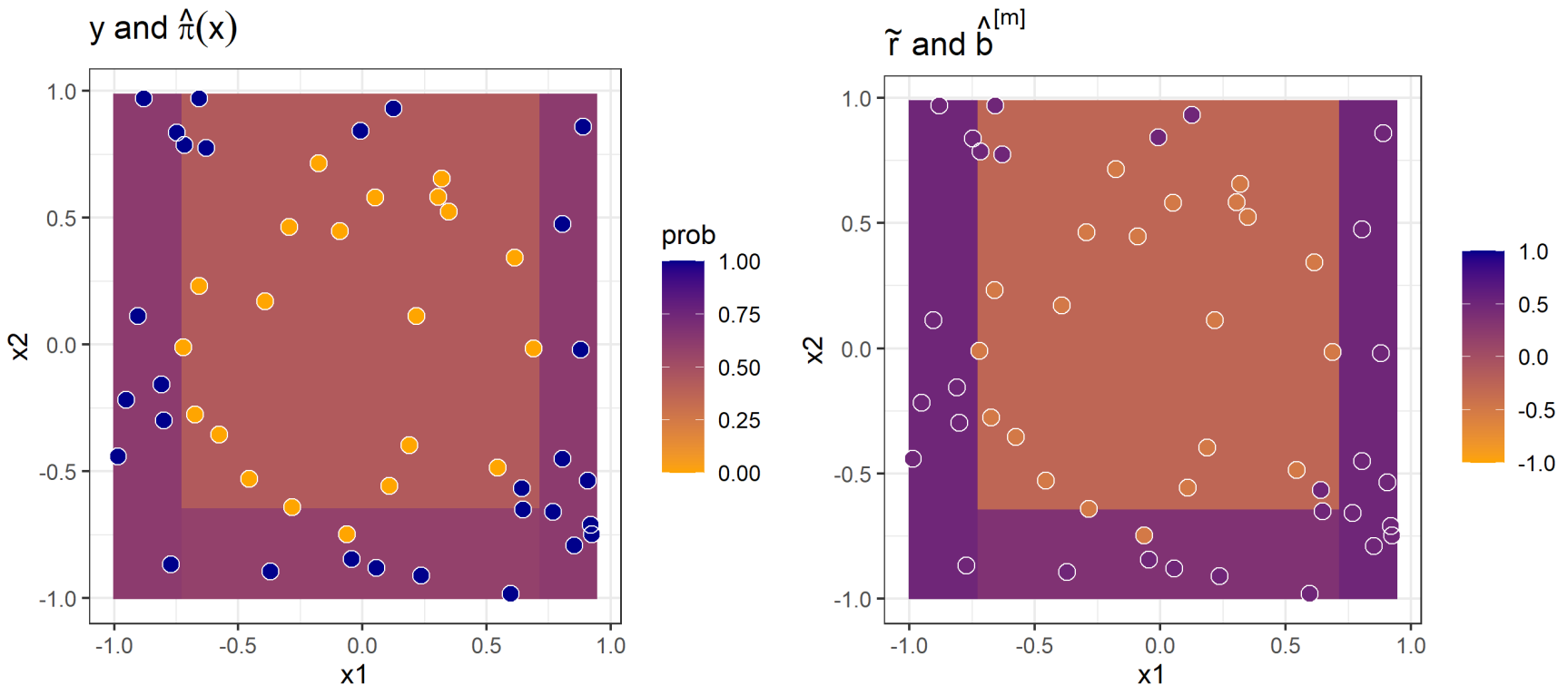
We now illustrate the boosting iterations for a classification example in a similar manner as we did for regression. However, we will now look at a simulation example with 2 instead of 1 influencial features and one binary target variable.

- We used the $\mathrm{mlbench}$ dataset circle with $n = 50$ observations.
- We used the Bernoulli loss to calculate pseudo-residuals and fitted in each iteration a base learner (regression tree with max. depth of 3) on them.
- We initialized with $f^{[0]} = log(1) = 0$.



EXAMPLE

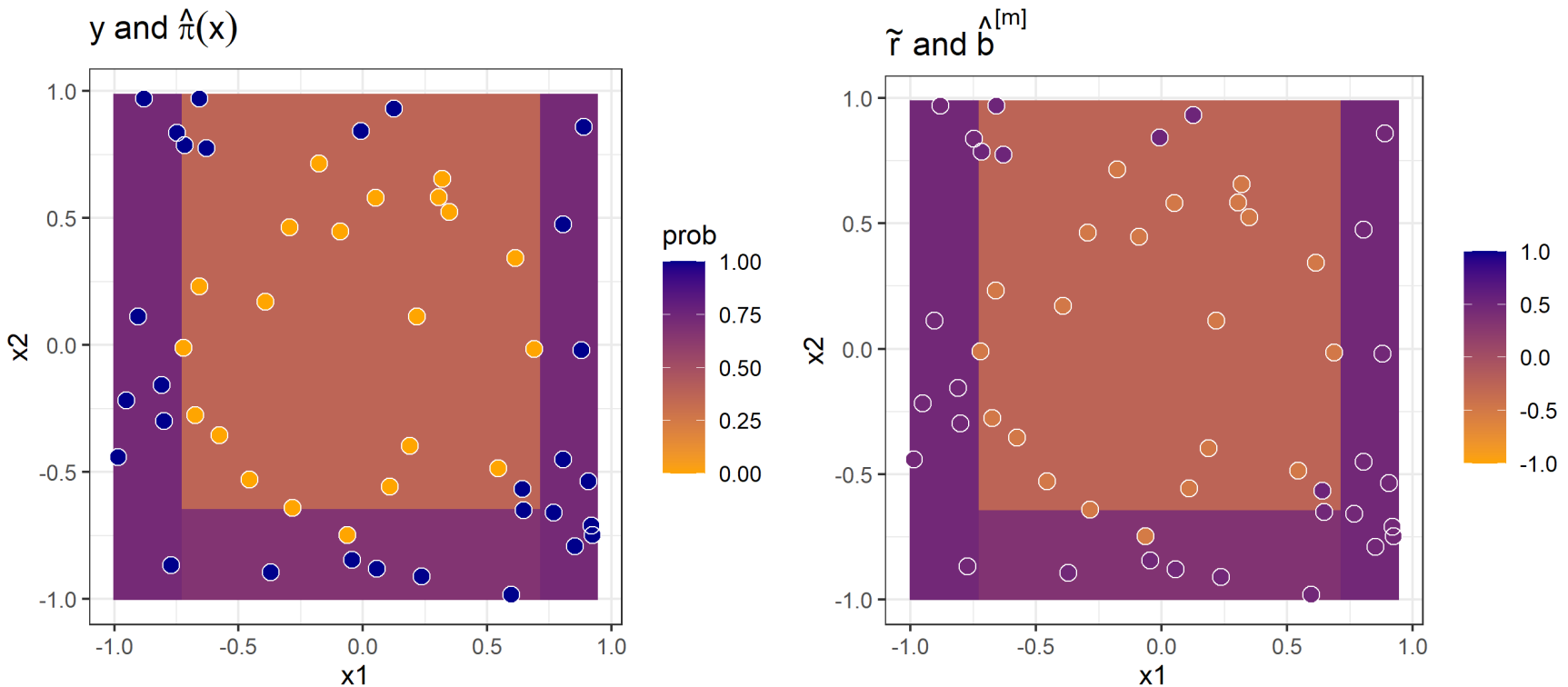
Left: Background color refers to prediction probabilites $\hat{\pi}^{[m]}$ and points to y (probabilities); Right: Background color refers to predictions of base learner $\hat{b}^{[m]}$ and points to $\tilde{r}$.



Iteration 1

EXAMPLE

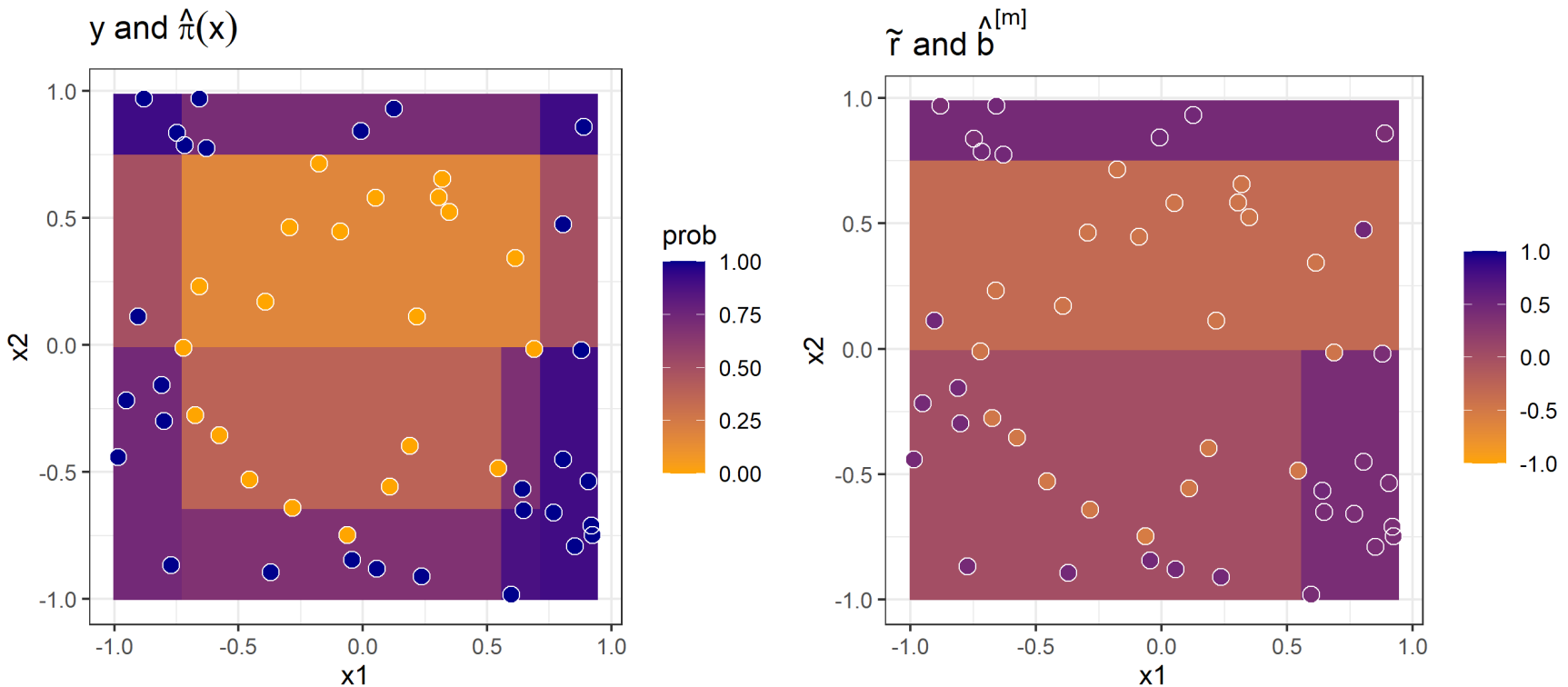
Left: Background color refers to prediction probabilites $\hat{\pi}^{[m]}$ and points to y (probabilities); Right: Background color refers to predictions of base learner $\hat{b}^{[m]}$ and points to $\tilde{r}$.



Iteration 2

EXAMPLE

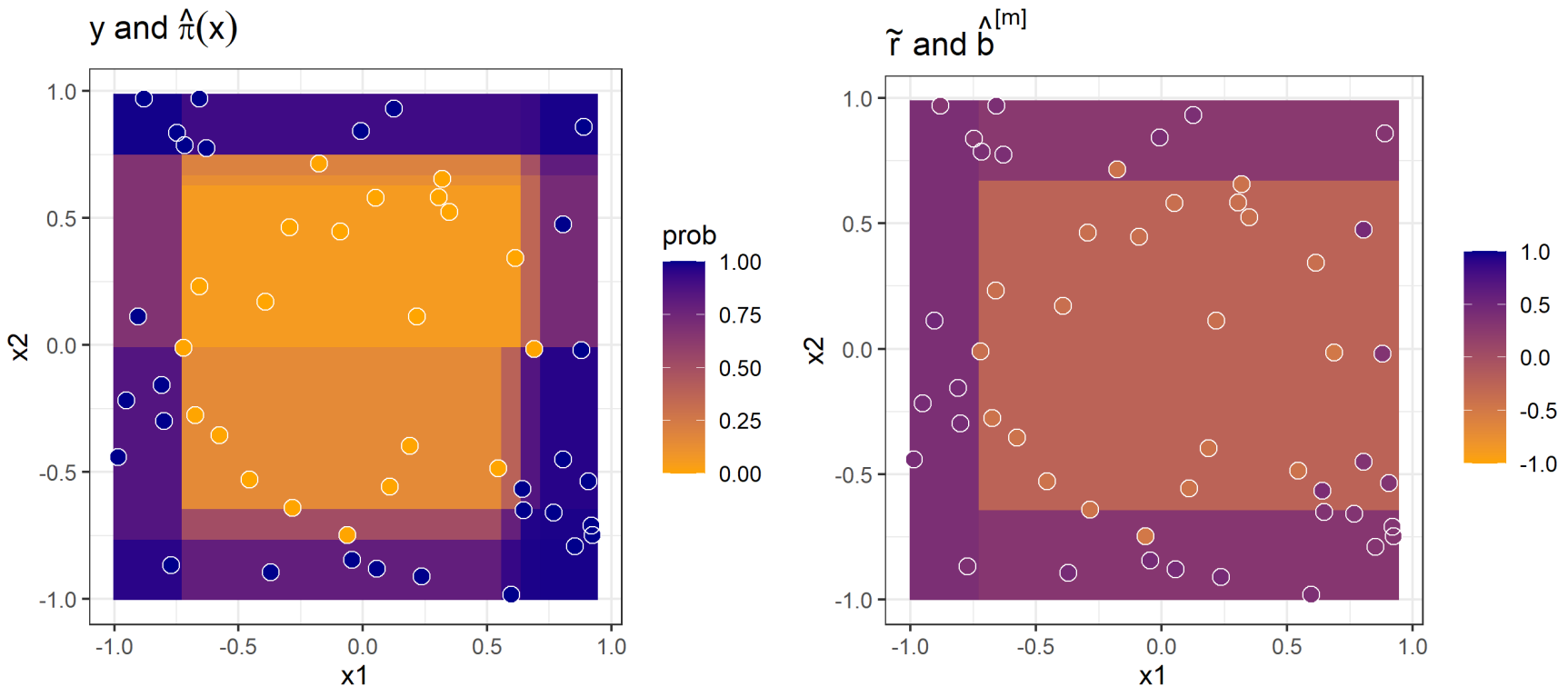
Left: Background color refers to prediction probabilites $\hat{\pi}^{[m]}$ and points to y (probabilities); Right: Background color refers to predictions of base learner $\hat{b}^{[m]}$ and points to $\tilde{r}$.



Iteration 5

EXAMPLE

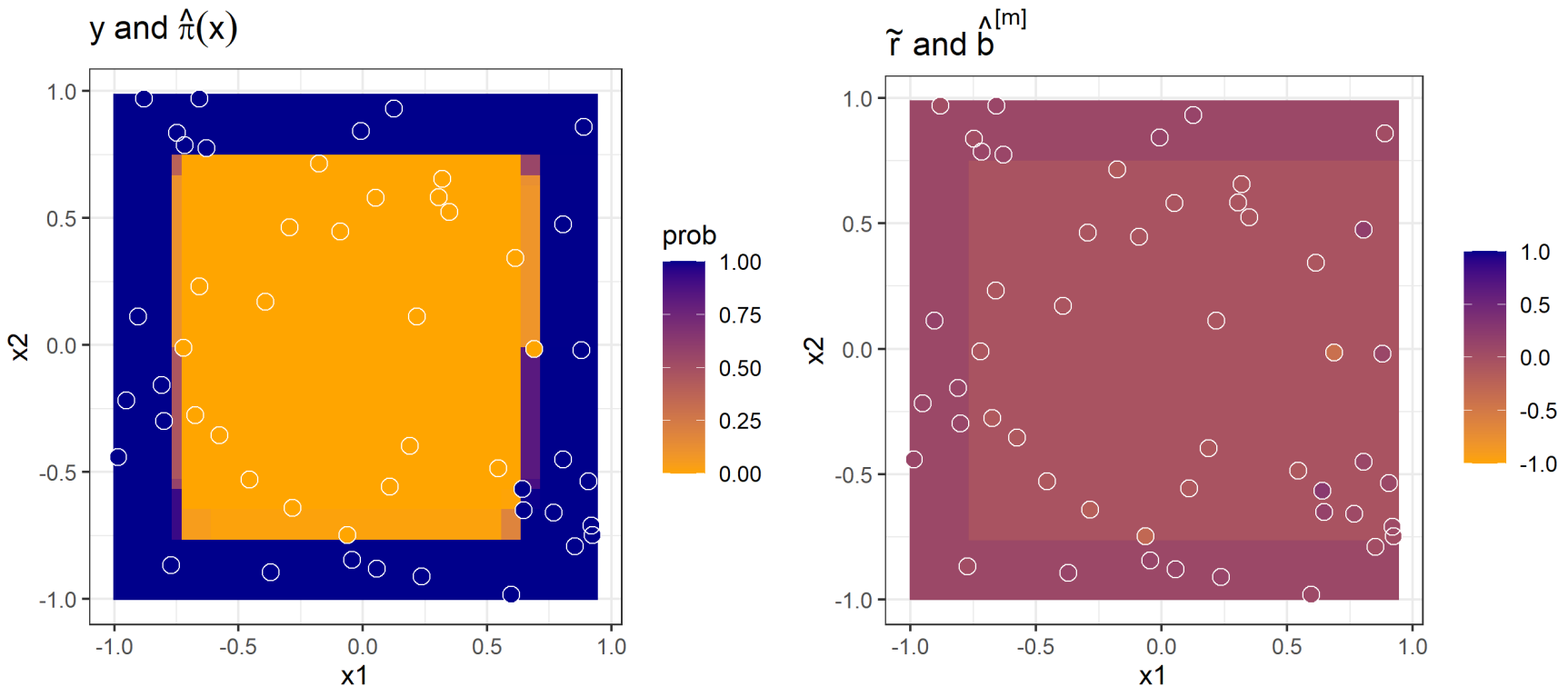
Left: Background color refers to prediction probabilites $\hat{\pi}^{[m]}$ and points to y (probabilities); Right: Background color refers to predictions of base learner $\hat{b}^{[m]}$ and points to $\tilde{r}$.



Iteration 10

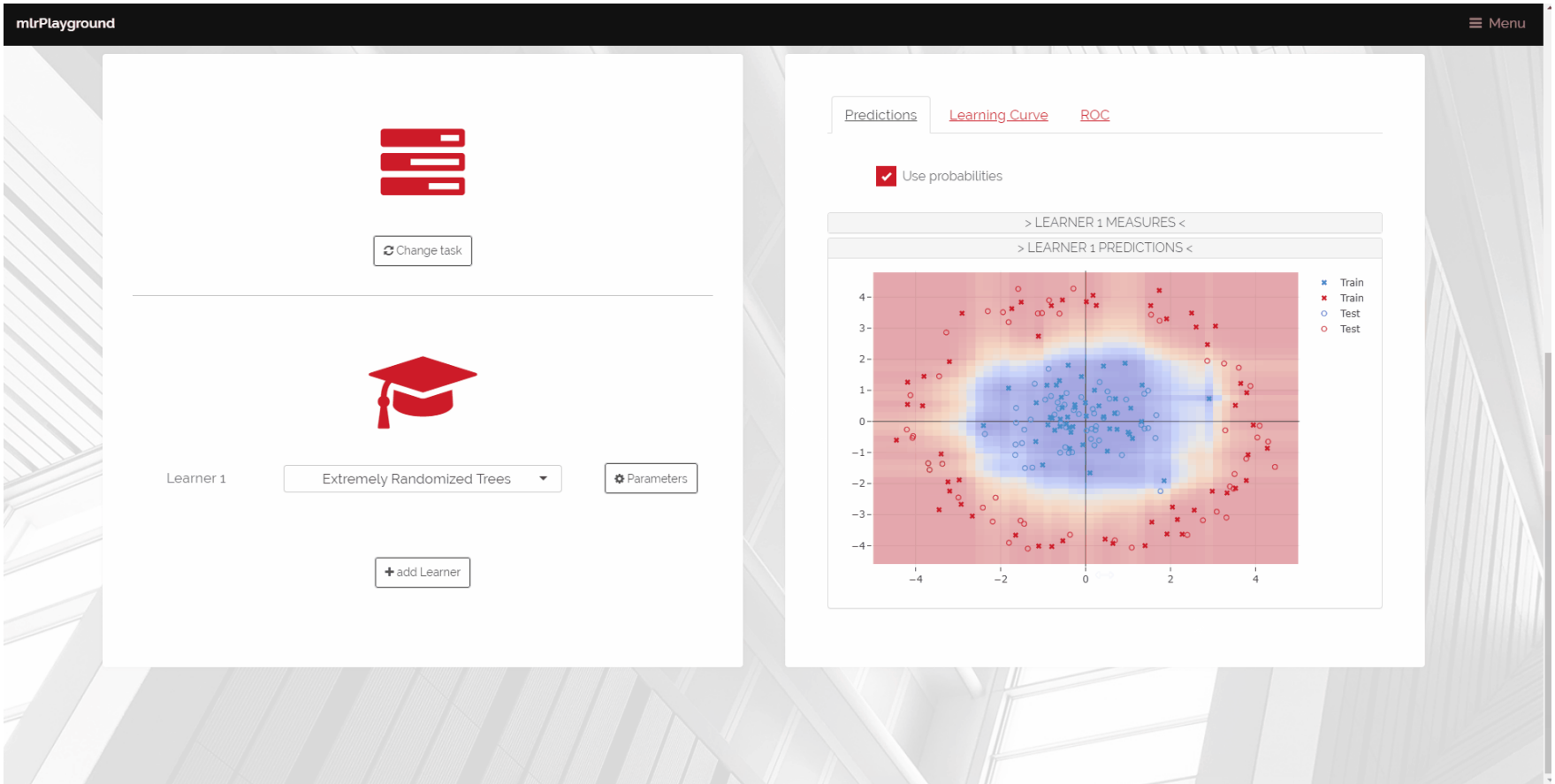
EXAMPLE

Left: Background color refers to prediction probabilites $\hat{\pi}^{[m]}$ and points to y (probabilities); Right: Background color refers to predictions of base learner $\hat{b}^{[m]}$ and points to $\tilde{r}$.



Iteration 100

MLRPLAYGROUND



▸ Open in browser.

MULTICLASS PROBLEMS

We proceed as in softmax regression and model a categorical distribution with multinomial / log loss. For $\mathcal{Y} = \{1, \ldots, g\}$, we create g discriminant functions $f_k(x)$, one for each class and each one being an **additive** model of base learners.
We define the $\pi_k(\mathbf{x})$ through the softmax function:

$$\pi_k(\mathbf{x}) = s_k(f_1(\mathbf{x}), \ldots, f_g(\mathbf{x})) = \exp(f_k(x)) / \sum_{j=1}^{g} \exp(f_j(\mathbf{x})).$$

Multinomial loss L:

$$L(y, f_1(\mathbf{x}), \ldots f_g(\mathbf{x})) = -\sum_{k=1}^{g} \mathbb{1}_{\{y=k\}} \ln \pi_k(\mathbf{x}).$$

Pseudo-residuals:

$$-\frac{\partial L(y, f_1(\mathbf{x}), \ldots, f_g(\mathbf{x}))}{\partial f_k(x)} = \mathbb{1}_{\{y=k\}} - \pi_k(\mathbf{x}).$$

MULTICLASS PROBLEMS

Algorithm GB for Multiclass

Initialize $f_k^{[0]}(\mathbf{x}) = 0,\ k = 1, \ldots, g$
for $m = 1 \to M$ **do**
    Set $\pi_k^{[m]}(\mathbf{x}) = \frac{\exp(f_k^{[m]}(\mathbf{x}))}{\sum_j \exp(f_j^{[m]}(\mathbf{x}))}, k = 1, \ldots, g$
    for $k = 1 \to g$ **do**
        For all i: Compute $\tilde{r}_k^{[m](i)} = \mathbb{1}_{\{y^{(i)}=k\}} - \pi_k^{[m]}(\mathbf{x}^{(i)})$
        Fit a regression base learner $\hat{b}_k^{[m]}$ to the pseudo-residuals $\tilde{r}_k^{[m](i)}$.
        Obtain $\hat{\beta}_k^{[m]}$ by constant learning rate or line-search.
        Update $\hat{f}_k^{[m]} = \hat{f}_k^{[m-1]} + \hat{\beta}_k^{[m]} \hat{b}_k^{[m]}$
    end for
end for
Output $\hat{f}_1^{[M]}, \ldots, \hat{f}_g^{[M]}$