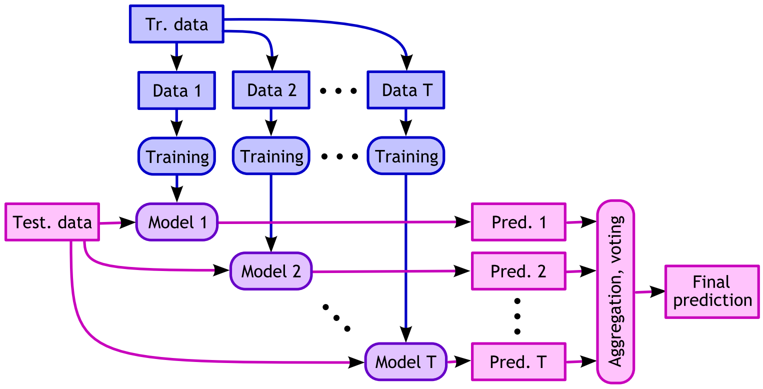
Boosting. Adaboost.

Petr Pošík

Ensembles, committees

Ensemble is a committee of several different models; their predictions are aggregated e.g. by voting or weighting.



Individual ensamble methods differ in the way they create individual *models different from each other*.

Boosting

Hypothesis Boosting Problem

- If there exists an efficient algorithm able to create *weak classifiers* (i.e. classifiers only slightly better than random guessing), does it also mean that there is an efficient algorithm able to build *strong classifiers* (i.e. classifiers with an arbitrary precision)?

Boosting algorithms

- iteratively learn weak classifiers using weighted training set,
- construct the final strong classifier as a weighted sum of the weak classifiers,
- assign the weights to individual weak learners depending on their accuracy,
- re-weight the training data for another round of the weak learner,
- differ in the way how they weight the training data and/or the individual weak classifiers.

AdaBoost

- Training data:
  - In each iteration $t = 1, \ldots, T$, it uses different weights $D_t(i)$ of the training examples x_i.
  - Incorrectly classified examples get a larger weight for the next iteration.
- The resulting classifier:
  - Weighted voting.

AdaBoost

Algorithm 1: AdaBoost

Input: Training set of labeled examples: $\{x_i, y_i\}, x_i \in \mathcal{R}^D, y_i \in \{+1, -1\}, i = 1, \ldots, m$

Output: Final classifier $H_{\text{final}}(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

begin

Initialize the weights of training examples: $D_1(i) = \frac{1}{m}$.

for $t = 1, \ldots, T$ **do**

Train a weak classifier h_t.

Compute the weighted error:

$$\epsilon_t = \sum_{i=1}^{m} D_t(i) I(y_i \neq h_t(x_i))$$

Compute the weight of classifier h_t:

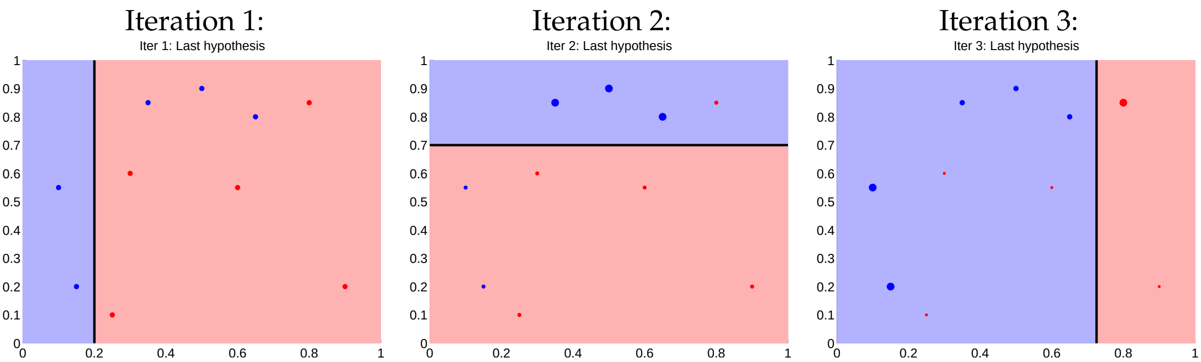
$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$$

Update the weights of the training examples:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } y_i = h_t(x_i), \\ e^{\alpha_t}, & \text{if } y_i \neq h_t(x_i), \end{cases}$$
$$= \frac{D_t(i)}{Z_t} \times \exp(-\alpha_t y_i h_t(x_i)),$$

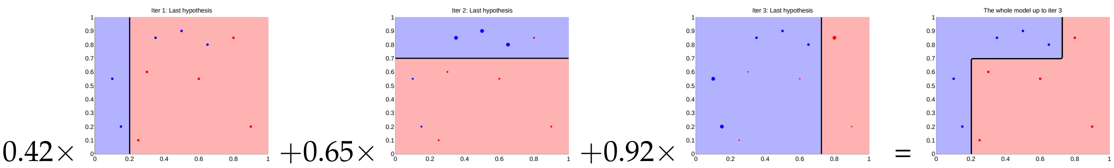
where Z_t is a normalization factor.

AdaBoost graphically



$\epsilon_1 = 0.3$ $\epsilon_2 = 0.21$ $\epsilon_3 = 0.13$

$\alpha_1 = 0.42$ $\alpha_2 = 0.65$ $\alpha_3 = 0.92$



AdaBoost: remarks

The training error:

- Let $\gamma_t = 0.5 - \epsilon_t$ be the improvement of the t-th model over a random guess.
- Let $\gamma = \min_t \gamma_t$ be the minimal improvement, i.e. the difference of error of all models $h(t)$ compared to the error of random guessing is at least γ, i.e.

$$\forall t : \gamma_t \geq \gamma > 0.$$

- It can be shown that the training error

$$\mathrm{Err}_{\mathrm{Tr}}(H_{\mathrm{final}}) \leq e^{-2\gamma^2 T}$$