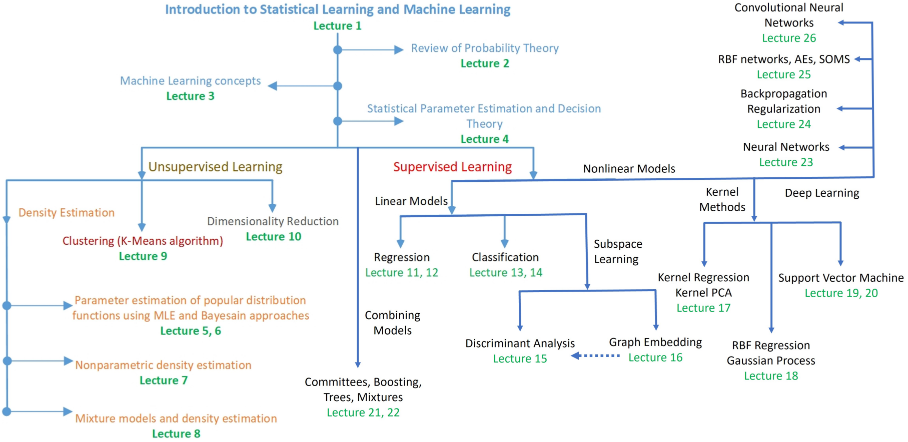
Statistical Learning and Machine Learning

Lecture 21 - Combining Models 1

October 13, 2021

Course overview and where do we stand



Why to combine multiple models?

Sometimes combining multiple models can lead to better performance compared to using only one:

- *Committees*: Use L different models and then make predictions using the average of the predictions of these L models
- *Boosting*: Use multiple models *in a sequence* in which each model's training is depending on the models preceding it in the sequence
- *Decision trees*: Divide the space in multiple regions and train one model for each region.
- *Mixture of experts*: Train K models and combine them based on a probabilistic mixture of the form:

$$p(t|\mathrm{x}) = \sum_{k=1}^{K} p(k|\mathrm{x}) p(t|\mathrm{x}, k) \tag{1}$$

Committees

Given a set of data points and their labels:

- We sample M subsets independently and we train a model on each of these subsets $y_m(\mathrm{x})$
- The committee prediction is:

$$y_{COM} = \frac{1}{M} \sum_{m=1}^{M} y_m(\mathrm{x}) \tag{2}$$

This procedure is called bootstrap aggregation or *bagging*.

Committees

Suppose the function we want to predict is $h(\mathrm{w})$, then:

$$y_m(\mathrm{x}) = h(\mathrm{x}) + \epsilon_m(\mathrm{x}) \tag{3}$$

The average sum-of-squares error has the form:

$$\mathbb{E}_{\mathrm{x}}[(y_m(\mathrm{x}) - h(\mathrm{x}))^2] = \mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathrm{x})^2] \tag{4}$$

So, the average error by the models acting individually is:

$$E_{AV} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathrm{x}}[\epsilon_m(\mathrm{x})^2] \tag{5}$$

The expected error of the committee is:

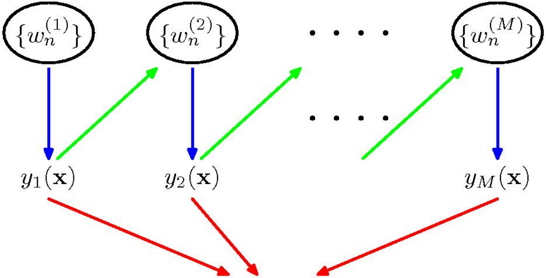
$$\mathbb{E}_{\mathrm{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} y_m(\mathrm{x}) - h(\mathrm{x})\right)^2\right] = \mathbb{E}_{\mathrm{x}}\left[\left(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\mathrm{x})\right)^2\right] \tag{6}$$

For zero-mean and uncorrelated errors, we obtain: $E_{COM} = \frac{1}{M} E_{AV}$.

AdaBoost

Boosting:

- Combination of multiple classifiers in a sequence
- The *base* classifiers do not need to be highly performing (*weak classifiers*)
- The combination of multiple weak classifiers leads to a high-performing committee of classifiers



$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

AdaBoost

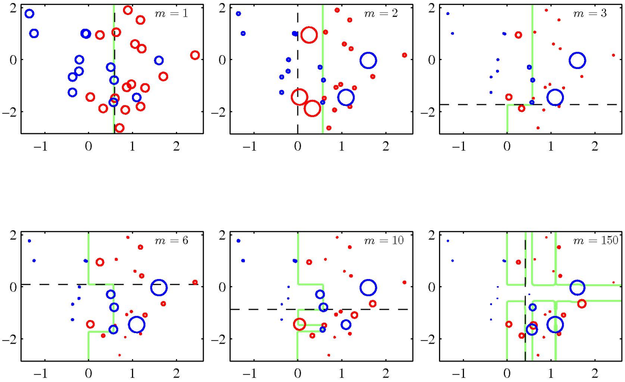
AdaBoost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.
2. For $m = 1, \ldots, M$:
   (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function
   $$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$
   where $I(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.

   (b) Evaluate the quantities
   $$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$
   and then use these to evaluate
   $$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}.$$

   (c) Update the data weighting coefficients
   $$w_n^{(m+1)} = w_n^{(m)} \exp \{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$
3. Make predictions using the final model, which is given by
$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x}) \right).$$

AdaBoost



Dashed black line: decision line of the most recent learner
Solid green line: combined decision line of the ensemble
Radius of points: weight assigned for training the most recent learner

AdaBoost

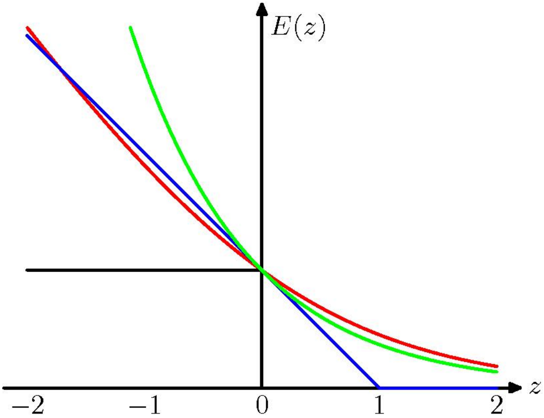
Error function for AdaBoost:

$$E = \sum_{n=1}^{N} exp\big(- t_n f_m(\mathrm{x}_n)\big) \tag{7}$$

where $t_n \in \{-1, 1\}$ and:

$$f_m(\mathrm{x}) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l y_l(\mathrm{x}) \tag{8}$$

AdaBoost



Green: Exponential function
Red: Cross-entropy function
Blue: Hinge error (used in SVM)
Black: misclassification error