

COMS30035, Machine learning: Combining Models 2, Ensembles

Edwin Simpson

`edwin.simpson@bristol.ac.uk`

Department of Computer Science, SCEEM
University of Bristol

October 30, 2020

Agenda

- ▶ Model Selection
- ▶ Model Averaging
- ▶ **Ensembles: Bootstrapping and Bagging**
- ▶ Tree-based Models
- ▶ Conditional Mixture Models
- ▶ Crowdsourcing

Ensemble Methods

- ▶ The combination of models performs better than the average individual, and often much much better than the best individual.
- ▶ Difference to BMA, where the weighted sum expresses uncertainty about which model is correct and tends to a single model as the dataset grows
- ▶ Difference to mixture of experts or decision trees: we don't split the input space between models, multiple models are combined for each data point.

Expected Error of an Ensemble

- ▶ Given a set of models, $1, \dots, M$, a simple combination the mean of the individual predictions, $y_{COM} = \frac{1}{M} \sum_{m=1}^M y_m(\mathbf{x})$, where $y_m(\mathbf{x}_i)$ is the prediction from model m .
- ▶ Let's compare the sum-of-squares error of y_{COM} with that of the individual models...
- ▶ Firstly, the expected error of our combination is:

$$E_{COM} = \mathbb{E}_{\mathbf{x}}[(y(\mathbf{x}) - y_{AV}(\mathbf{x}))^2] = \mathbb{E}_{\mathbf{x}} \left[\left(\frac{1}{M} \sum_{m=1}^M y(\mathbf{x}) - y_m(\mathbf{x}) \right)^2 \right]. \quad (1)$$

Expected Error of an Ensemble

- ▶ $E_{COM} = \mathbb{E}_{\mathbf{x}} \left[\left(\frac{1}{M} \sum_{m=1}^M y(\mathbf{x}) - y_m(\mathbf{x}) \right)^2 \right].$
- ▶ The average error of an individual model is:
 $E_{AV} = \frac{1}{M} \sum_{m=1}^M \mathbb{E}[(y(\mathbf{x}) - y_m(\mathbf{x}))^2].$
- ▶ If we make two assumptions...
 1. The errors of each model have zero mean;
 2. The errors of different models are not correlated;
- ▶ ...then we arrive at $E_{COM} = \frac{1}{M} E_{AV}!$
- ▶ Intuition: if models make different, random errors, they will tend to cancel out.

Expected Error of an Ensemble

- ▶ $E_{COM} = \frac{1}{M}E_{AV}$ is pretty amazing, but is it realistic?
- ▶ No, because we have made extreme assumptions about the models' errors – in practice, they are usually highly correlated and biased.
- ▶ However, the combined error cannot be worse than the average error: $E_{COM} \leq E_{AV}$ ¹
- ▶ The results tells us that the models should be *diverse* to avoid repeating the same errors.

¹This bound is due to *Jensen's inequality*.

Bootstrap Aggregation (Bagging)

- ▶ Bagging is a simple ensemble method that induces diversity by training M models on different samples of the training set.
- ▶ For each model m , randomly sample N data points with replacement from a training set with N data points.
- ▶ In each bootstrap dataset, some data points will be repeated and others will be omitted.
- ▶ Combine predictions by taking the mean.

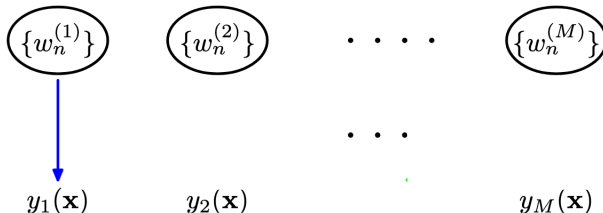
Boosting

- ▶ Can we do better than choosing training sets at random?
- ▶ We train the *base* models in sequence to ensure that each base model addresses the weaknesses of the ensemble.
- ▶ Instead of training a new base model on a random sample, weight the data points in the training set according to the performance of previous base models.
- ▶ *AdaBoost* is a popular boosting method for *binary classification*.

AdaBoost

Training sequence \rightarrow

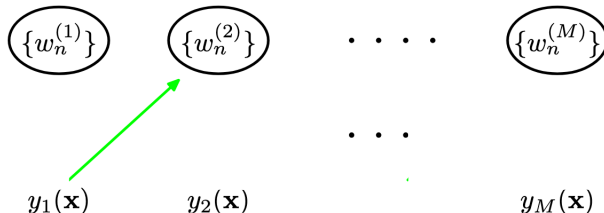
train new classifier
on weighted data
that outputs class
labels $+1$ or -1



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

AdaBoost

Training sequence \rightarrow

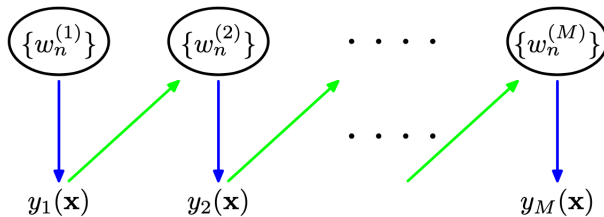


compute weights
from performance of
previous classifier

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

AdaBoost

Training sequence \rightarrow



after all classifiers are trained, combine using a weighted sum

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

AdaBoost: Data Weights

1. Initialise to $1/N$;
2. Compute the weighted accuracy of model m :

$$\epsilon_m = \sum_{n=1}^N w_n^{(m)} [y_m(\mathbf{x}_n) \neq y(\mathbf{x}_n)] / \sum_{n=1}^N w_n^{(m)} \quad (2)$$

3. Update the weight for each data point n :

$$w_n^{(m+1)} = \begin{cases} w_n^{(m)} \left(\frac{1-\epsilon_m}{\epsilon_m} \right) & \text{if } y_m(\mathbf{x}_n) \neq y(\mathbf{x}_n) \\ w_n^{(m)} & \text{if } y_m(\mathbf{x}_n) = y(\mathbf{x}_n) \end{cases} \quad (3)$$

- The weight is increased when m makes an incorrect prediction.

AdaBoost: Final Classifier Weights

- ▶ AdaBoost chooses weight the α_m for m that minimises the exponential loss of base classifier m given previous classifiers:

$$y_M(\mathbf{x}_n) = \sum_{m=1}^M \alpha_m y_m(\mathbf{x}_n), \quad (4)$$

$$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\} \quad (5)$$

- ▶ Weights are higher for classifiers with a lower error rate.
- ▶ Note that α_m is a log-odds function: AdaBoost optimises the approximation to the log-odds ratio.
- ▶ Other loss functions can be used to derive similar boosting schemes for regression and multi-class classification.

Stacking

Edwin Simpson

edwin.simpson@bristol.ac.uk

COMS30035, Machine learning: Combining Models 2, Ensembles

Now do the quiz!

Please do the quiz for this lecture on Blackboard.