CS-667 Advanced Machine Learning

Nazar Khan

PUCIT

Boosting

Combining Models

- So far, we have seen many models for classification and regression.
- Often, using a combination of models is better than a single model.
- Combinations of models are called *committees*.
- A simple committee can be obtained by training M models and then taking the average of their predictions.

$$y_{\text{COM}}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x})$$

Bagging

- Train M models on M subsets of data.
- The subsets are called *bootstrap* datasets.
- This introduces variability among the learned models.
- Prediction of committee is as before by averaging (*i.e.*, aggregating the models and dividing by M).
- Hence the name *bootstrap aggregation* or *bagging*.

Boosting

- A highly successful technique for combining models is called *boosting*.
- Originally devised for classification problems.
- Basic idea is to obtain a strong learner by combining the outputs of multiple weak learners.
- Difference from standard model averaging and bagging is that the models are trained *in sequence*.
- Misclassified points are given more importance when training subsequent classifiers.
- By combining the models in an adaptive fashion, we obtain the *adaptive boosting* or *AdaBoost* algorithm.
  - In the final combined model, each individual classifier is weighted by how well it performed on the training data.

The AdaBoost Algorithm

Given data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and labels $t_1, \dots, t_N$.
Goal is to learn a final classifier Y_M made up from base classifiers $y_1, \dots, y_M$.

1. Initialise data point weights $w_n^{(1)} = \frac{1}{N}$ for $n = 1, \dots, N$.
2. For $m = 1, \dots, M$
   - 2.1 Fit a weighted model y_m to the data using weights $w_n^{(m)}$.
   - 2.2 Evaluate

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} \mathbb{I}(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

where $\mathbb{I}(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function that equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.
Notice that $0 \leq \epsilon_m \leq 1$ (0 for all correct, 1 for all misclassified). So it determines how good model y_m is.

The AdaBoost Algorithm

2.3 Evaluate

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

which is large when model y_m is good.

2.4 Update data point weights

$$w_n^{(m+1)} = \begin{cases} w_n^{(m)} & \text{if } \mathbf{x}_n \text{ is correctly classified} \\ w_n^{(m)}\frac{1-\epsilon_m}{\epsilon_m} & \text{if } \mathbf{x}_n \text{ is misclassified} \end{cases}$$

3. Construct final model as

$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)$$