Lecture 23

Name: Harsh Sanjay Roniyar

Roll Number: 22B3942

- Ensemble
- Bagging
- Cascade (Tree)
- Boosting

Assumption: Weak Learners

Adaboost

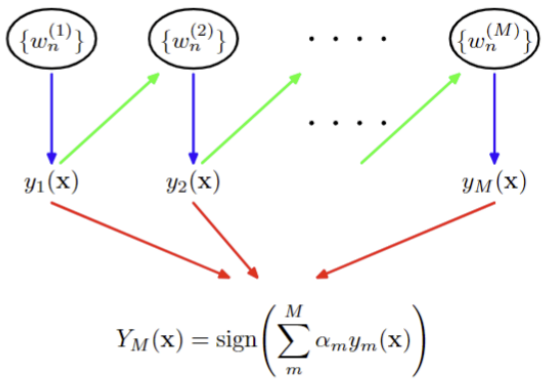

Figure 14.1 Schematic illustration of the boosting framework. Each base classifier $y_m(\mathbf{x})$ is trained on a weighted form of the training set (blue arrows) in which the weights $w_n^{(m)}$ depend on the performance of the previous base classifier $y_{m-1}(\mathbf{x})$ (green arrows). Once all base classifiers have been trained, they are combined to give the final classifier $Y_M(\mathbf{x})$ (red arrows).

> Christopher M. Bishop - Pattern Recognition and Machine Learning-Springer (2011), page 627

**The Algorithm:

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = \frac{1}{N}$ 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

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

   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.

   2. Evaluate the quantities

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

and then use these to evaluate the model weights (`assumption:` $\epsilon < \frac{1}{2}$ `i.e. a weak learner`)

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

3. Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{\alpha_m \mathbb{1}(y_m(\mathbf{x}_n) \neq t_n)\right\}.$$

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 is minimizing exponential error.

Overall Loss:

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x_n})\}$$

where, $f_m(\mathbf{x})$ is the weighted average given by -

$$f_m(\mathbb{x}) = \frac{1}{2}\sum_{l=1}^{m} \alpha_l y_l(\mathbf{x})$$

Also,

$$w_n^{(m)} = \exp\{-t_n f_{m-1}(\mathbf{x_n})\}$$

and thus, E reduces to

$$E = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x_n})\right\}$$

T_m: correctly classified
M_m: misclassified subset

$$\boxed{w_n^{(m+1)} = w_n^{(m)} \exp(-\alpha_m/2) \exp\left\{\alpha_m \mathbb{1}(y_m(\mathbf{x_n}) \neq t_n)\right\}}$$

Unsupervised Learning

Input: D-dim continuous

1. Clustering (Output: Discrete Brackets)
2. Dimension Reduction (Output: Continuous Numbers)

Principle:

1. Reduce data precision required in the output
2. Still be able to reconstruct the input from output.

Error Function: Reduce the reconstruction loss of the L2 Norm

Clustering

- Identify if there are any naturally visible groups, sometimes it might not be obvious and hence need mathematical ways.
- Clustering can also help in reducing data, by using only a cluster descriptor parameter for the data-points belonging to the cluster.

K-means clustering

Divides the region into K-hard partitions.

Input: $X \equiv \{x_1, x_2, \ldots, x_N\}$, K

Initialize: $c_1, c_2, \ldots, c_K$

Loop
{
$y_n \leftarrow \arg\min_k d(x_n, c_k)$
$c_k \leftarrow \frac{\sum_{n=1}^{N} \mathbb{1}\{y_n=k\} x_n}{\sum_{n=1}^{N} \mathbb{1}\{y_n=k\}}$ for all n, k
}
Until no change in cluster assignment

Loss: $\sum_n \sum_k \mathbb{1}\{y_n = k\} \|x_n - c_k\|_2^2$