Model ensembles

Tim Lawson

November 17, 2023

Model ensembles

- Learn multiple models from versions of the data
  - Resample, e.g., bagging, subspace sampling
  - Reweight, e.g., boosting
- Combine the outputs of the models
  - Average scores or probabilities
  - Majority vote

Boosting

Definition

- $\vec{x}_i \in \mathbb{R}^n$ is an instance
- $\vec{y}_i \in \{0, 1\}^k$ is a label (one-hot vector)
- $f^{(j)} : \mathbb{R}^n \rightarrow \{0, 1\}^k$ is a model
- $\hat{\vec{y}}_i^{(j)} = f^{(j)}(\vec{x}_i) \in \{0, 1\}^k$ is a prediction (one-hot vector)
- $w_i^{(j)} \in \mathbb{R}, w_i^{(0)} = \frac{1}{n}, \sum_{i=1}^n w_i^{(j)} = 1$ is an instance weight
- $\epsilon^{(j)} = \sum_{i:\hat{\vec{y}}_i^{(j)} \neq \vec{y}_i} w_i^{(j)} \in \mathbb{R}$ is the weighted error of model $f^{(j)}$
- $\alpha^{(j)} = f_\alpha(\epsilon^{(j)}) \in \mathbb{R}$ is the weight of model $f^{(j)}$
- $w_i^{(j+1)} = f_w(w_i^{(j)}, \vec{y}_i, \hat{\vec{y}}_i^{(j)}, \epsilon^{(j)})$ is the updated instance weight
- $\hat{\vec{y}}_i = \sum_{j=1}^J \alpha^{(j)} f^{(j)}(x_i) \in \{0, 1\}^k$ is the ensemble model prediction

Boosting
Code

```
def get_model_weight(weighted_error: float) -> float:
    """Get the weight of a model."""
    raise NotImplementedError


def update_weight(
    weight: float,
    label: int,
    prediction: int,
    weighted_error: float,
) -> float:
    """Update the weight of an instance."""
    raise NotImplementedError
```

Boosting

Questions

- What should the weights of the models f_α/get_model_weights be?
- What should the weight updates f_w/update_weight be?

Boosting

Model weights derivation

Assume that the weight updates f_w are:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z^{(j)}} \times \begin{cases} e^{-\alpha^{(j)}} & \text{if } \vec{\hat{y}}_i^{(j)} = \vec{y}_i \\ e^{\alpha^{(j)}} & \text{otherwise} \end{cases}$$

This can be simplified with:

$$\delta(\vec{y}_i, \vec{\hat{y}}_i^{(j)}) = \begin{cases} 1 & \text{if } \vec{\hat{y}}_i^{(j)} = \vec{y}_i \\ -1 & \text{otherwise} \end{cases}$$

$$w_i^{(j+1)} = w_i^{(j)} \frac{\exp(-\alpha^{(j)} \delta(\vec{y}_i, \vec{\hat{y}}_i^{(j)}))}{Z^{(j)}}$$

Boosting

Model weights derivation

Each update is multiplicative:

$$w_i^{(J)} = w_i^{(0)} \prod_{j=1}^{J} \frac{\exp(-\alpha^{(j)} \delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)}))}{Z^{(j)}} = \frac{1}{n} \frac{\exp(-\delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)}))}{\prod_{j=1}^{J} Z^{(j)}}$$

Each set of instance weights sums to 1:

$$1 = \sum_{i=1}^{n} w_i^{(j)} = \sum_{i=1}^{n} \frac{1}{n} \frac{\exp(-\delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)}))}{\prod_{j=1}^{J} Z^{(j)}}$$

$$\prod_{j=1}^{J} Z^{(j)} = \frac{1}{n} \sum_{i=1}^{n} \exp(-\delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)}))$$

$\exp(-\delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)})) \geq 1$ if x_i is misclassified by the ensemble, so $\prod_{j=1}^{J} Z^{(j)}$ is an upper bound on the ensemble error.

Boosting

Model weights derivation

$\prod_{j=1}^{J} Z^{(j)}$ could be minimized by minimizing the model error $(n)Z^{(j)}$:

$$nZ^{(j)} = \sum_{i=1}^{n} w_i^{(j)} \exp(-\alpha^{(j)} \delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)}))$$

By the definitions of $\epsilon^{(j)}$ and $\delta(\vec{y}_i, \hat{\vec{y}}_i^{(j)})$:

$$nZ^{(j)} = \epsilon^{(j)} \exp(\alpha^{(j)}) + (1 - \epsilon^{(j)}) \exp(-\alpha^{(j)})$$

Boosting

Model weights derivation

Therefore, $Z^{(j)}$ is minimized when:

$$\frac{\partial Z^{(j)}}{\partial \alpha^{(j)}} = \epsilon^{(j)} \exp(\alpha^{(j)}) - (1 - \epsilon^{(j)}) \exp(-\alpha^{(j)}) = 0$$

$$\exp(2\alpha^{(j)}) = \frac{1 - \epsilon^{(j)}}{\epsilon^{(j)}}$$

That is:

$$\alpha^{(j)} = \frac{1}{2} \ln \left(\frac{1 - \epsilon^{(j)}}{\epsilon^{(j)}} \right), \;\; Z^{(j)} = 2\sqrt{\epsilon^{(j)}(1 - \epsilon^{(j)})}$$