

Machine Learning Using Python

Lesson 7: Model Stacking

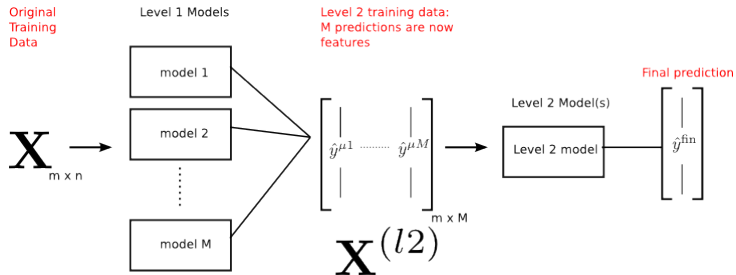
Marcel Scharth

The University of Sydney Business School

Model stacking

- **Model stacking** is a learning method that aims to improve predictive accuracy by combining predictions from multiple models. It is a particular case of **ensemble learning**.
- Model stacking will usually lead to better predictions, at the expense of interpretability. It is a component of several winning entries in Kaggle competitions.

Model Stacking



Source: <https://www.kdnuggets.com/2017/02/stacking-models-improved-predictions.html>

Model averaging

Consider the regression setting and suppose that we estimate M models $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_M$, leading to corresponding predictions $\hat{f}_1(\mathbf{x}), \dots, \hat{f}_M(\mathbf{x})$.

In **Model averaging**, we compute the prediction

$$\hat{f}_{\text{ave}}(\mathbf{x}) = \sum_{m=1}^M w_m \hat{f}_m(\mathbf{x}),$$

where w_1, \dots, w_m are fixed model weights.

Model averaging

How to choose the weights? One option is to simply pick the model weights, say $w_m = 1/M$ for a simple model average. Or we can choose the coefficients by least squares,

$$\min_{w_1, \dots, w_m} \sum_{i=1}^N \left(y_i - \sum_{m=1}^M w_m \hat{f}_m(\mathbf{x}_i) \right)^2,$$

where we may want to impose restrictions on the weights such as non-negativity and a total sum of one.

Model averaging

However, least squares estimation

$$\min_{w_1, \dots, w_M} \sum_{i=1}^N \left(y_i - \sum_{m=1}^M w_m \hat{f}_m(\mathbf{x}_i) \right)^2,$$

may not work well since the regression models $\hat{f}_1(\cdot), \dots, \hat{f}_M(\cdot)$ were previously estimated based on the same training data.

The minimisation does not take into account the complexity of the individual models, and will tend to overfit by putting too much weight on the most complex models (which will have low training errors).

Model stacking

Stacking overcomes this difficulty by solving the minimisation problem

$$\min_{w_1, \dots, w_M} \sum_{i=1}^N \left(y_i - \sum_{m=1}^M w_m \hat{f}_m^{(cv)}(\mathbf{x}_i) \right)^2,$$

where $\hat{f}_m^{(cv)}(\mathbf{x}_i)$ are cross validation predictions.

Model stacking

In above case, model stacking fits a linear regression model

$$\min_{w_1, \dots, w_m} \sum_{i=1}^N \left(y_i - \sum_{m=1}^M w_m \hat{f}_m^{(cv)}(\mathbf{x}_i) \right)^2,$$

based on constructed predictors derived from different models. We say that the linear regression model is therefore the **meta model** for the stack.

Model stacking allows us generalise this idea and use any algorithm as a meta model.

Classification

Model averaging and stacking extends to classification. Some options are:

- Hard voting classifier: (weighted) majority voting across several models.
- Soft voting classifier: (weighted) probability averages.
- Model stacking: fit a meta model such as a logistic regression using cross validation predictions from multiple models as inputs.