Lecture #17: Stacking

Data Science 1

CS 109A, STAT 121A, AC 209A, E-109A

Pavlos Protopapas   Kevin Rader

Rahul Dave   Margo Levine

Lecture Outline

Review

Stacking

Review

Review of Ensemble Methods

So far we've seen a variety of ensemble methods:

- **Bagging**
  - simultaneous training using bootstrap samples of data but the same set of predictors
  - ensemble is averaged to produce final model
- **Random Forest**
  - simultaneous training using bootstrap samples of data
  - models trained on random samples of predictors
  - ensemble is averaged to produce final model
- **Boosting**
  - serial training using common set of data and predictors
  - each new model is trained focusing on regions of error in the previous model
  - ensemble is summed to produce the final model

Review of Ensemble Methods

So far we've seen a variety of ensemble methods:

- **Bagging and Random Forest**
  - low bias - ensemble of complex models
  - low variance - variance is reduced via averaging and de-correlating models in ensemble
- **Boosting**
  - low bias - training error iteratively reduced
  - low variance - ensemble of simple models

Stacking

Motivation for Stacking

Recall that in boosting, the final model T, we learn is a weighted sum of simple models, T_h,

$$T = \sum_h \lambda_h T_h.$$

where λ_h is the learning rate. In AdaBoost for example, we can analytically determine the optimal values of λ_h for each simple model T_h.

On the other hand, we can also determine the final model T implicitly by ***learning any model, called meta-learner, that transforms the outputs of T_h into a prediction***.

Stacked Generalization

The framework for ***stacked generalization*** or ***stacking*** (Wolpert 1992) is:

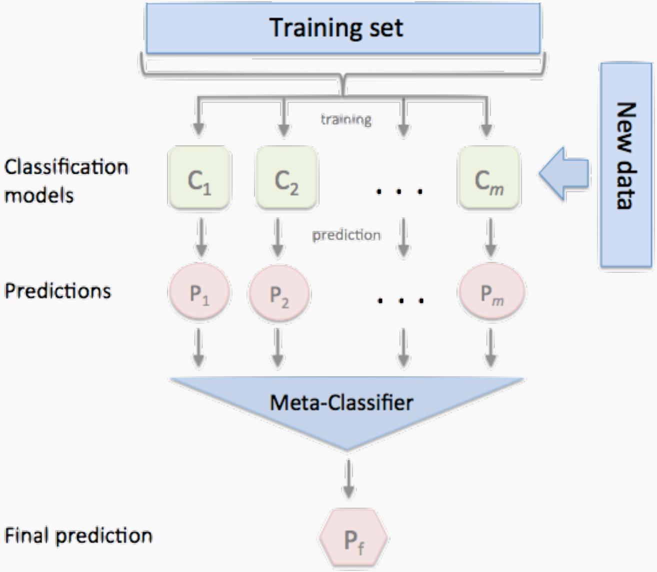
- train L number of models, T_l on the training data

$$\{(x_1, y_1), \ldots, (x_N, y_N)\}$$

- train a meta-learner $\widetilde{T}$ on the predictions of the ensemble of models, i.e. train using the data

$$\{(T_1(x_1), \ldots, T_L(x_1), y_1), \ldots, (T_1(x_N), \ldots, T_L(x_N), y_N)\}$$

Stacked Generalization



Stacked Generalization

Stacking is a very general method,

- the models, T_l, in the ensemble can come from different classes. The ensemble can contain a mixture of logistic regression models, trees etc.
- the meta-learner, T, can be of any type.

Note: we want to train T on the ***out of sample*** predictions of the ensemble. For example we train T on

$$\{(T_1(x_1), \ldots, T_L(x_1), y_1), \ldots, (T_1(x_N), \ldots, T_L(x_N), y_N)\}$$

where $T_l(x_n)$ is generated by training T_l on

$$\{(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (x_{n+1}, y_{n+1}), \ldots, (x_N, y_N)\}.$$

Stacking: General Guidelines

The flexibility of stacking makes it widely applicable but difficult to analyze theoretically. Some general rules have been found through empirical studies:

- models in the ensemble should be diverse, i.e. their errors should be uncorrelated
- for classification, each model in the ensemble should have error rate $< 1/2$
- if models in the ensemble outputs probabilities, it's better to train the meta-learner on probabilities rather than predictions
- apply regularization to the meta-learner to avoid overfitting

Stacking: Subsemble Approach

We can extend the stacking framework to include ensembles of models that specialize on small subsets of data (Sapp et. al. 2014), for de-correlation or improved computational efficiency:

- divide the data in to J subsets
- train models , T_j, on each subset
- train a meta-learner $\widetilde{T}$ on the predictions of the ensemble of models, i.e. train using the data

$$\{(T_1(x_1), \ldots, T_J(x_1), y_1), \ldots, (T_1(x_N), \ldots, T_J(x_N), y_N)\}$$

Again, we want to make sure that each $T_j(x_n)$ is an out of sample prediction.