Ensemble di modelli

Data Mining
CLAMSES - University of Milano-Bicocca

Aldo Solari

Riferimenti bibliografici

- HTF § 8.8
- Getting Started With `stacks`

Ensemble di modelli

Si consideri un *ensemble* di L modelli $\hat{f}_1, \ldots, \hat{f}_L$ stimati sui dati di training

$$(x_1, y_1), \ldots, (x_n, y_n)$$

L'idea è di combinare le previsioni di diversi modelli per ottenere una previsione migliore rispetto a quella di modelli individuali.

Il problema è di come combinare le previsioni per prevedere le risposte del test set $y_1^*, \ldots, y_m^*$

Model stacking

Stacking o *stacked generalization* è un metodo generale per combinare un insieme di modelli

Stacking considera la combinazione lineare

$$\hat{y}_i^* = \sum_{l=1}^{L} w_l \hat{f}_l(x_i^*)$$

e richiede di definire i pesi $w_1, \ldots, w_L$

Minimi quadrati

Il metodo dei minimi quadrati fornisce come soluzione

$$\hat{w}_1, \ldots, \hat{w}_L = \underset{w_1,\ldots,w_L}{\arg\min} \sum_{i=1}^{n} \left[y_i - \sum_{l=1}^{L} w_l \hat{f}_l(x_i) \right]^2$$

Tuttavia, in questo modo non si tiene conto della complessità del modello: i modelli con maggiore complessità ottengono pesi più elevati.

Per esempio, se consideriamo L predittori e definiamo $\hat{f}_l$ come il *best subset* di dimensione l, $l = 1, \ldots, L$ (ovvero il modello *best subset* con l predittori minimizza MSE_{Tr} rispetto agli altri modelli di uguale dimensione). In questo caso tutto il peso va sul modello più complesso, cioè $\hat{w}_L = 1$ and $\hat{w}_l = 0$ for $l < L$

Leave-one-out cross validation

Se escludiamo y_i nella procedura di stima del modello, allora $\hat{f}_1^{-i}(x_i), \ldots, \hat{f}_L^{-i}(x_i)$ non dipende da y_i:

$$\hat{w}_1, \ldots, \hat{w}_L = \underset{w_1,\ldots,w_L}{\arg\min} \sum_{i=1}^{n} \left[y_i - \sum_{l=1}^{L} w_l \hat{f}_l^{-i}(x_i) \right]^2$$

Stacked generalization (Wolpert, 1992; Breiman, 1996) è un metodo di *ensemble* dove il modello combinante è un modello lineare con pesi determinati via *leave-one-out cross validation*

Algoritmo *stacked generalization*

1. Let $\hat{f}_l^{-i}(x_i)$ be the prediction at x_i using model l fitted to the training data with the ith training observation (x_i, y_i) removed
2. Obtain the weights by least squares

$$\hat{w}_1, \ldots, \hat{w}_L = \underset{w_1,\ldots,w_L}{\arg\min} \sum_{i=1}^{n} \left[y_i - \sum_{l=1}^{L} w_l \hat{f}_l^{-i}(x_i) \right]^2$$

3. Compute the predictions for the test data as

$$\hat{f}_{\text{stack}}(x_i^*) = \sum_{l=1}^{L} \hat{w}_l \hat{f}_l(x_i^*), \quad i = 1, \ldots, m$$

Esercizio 1

Si considerino i dati `Boston` presenti nella libreria `MASS`. I dati di training e di test data sono (si veda la slide successiva)

$$(x_1, y_1), \ldots, (x_n, y_n), \qquad (x_1^*, y_1^*), \ldots, (x_m^*, y_m^*)$$

dove $n = 235$ e $m = 271$.

La variabile risposta è `medv`, mentre i predittori sono `crim`, `zn`, `indus`, `chas`, `nox`, `rm`, `age`, `dis`, `rad`, `tax`, `ptratio`, `black`, `lstat`

Si consideri un *ensemble* di $L = 2$ modelli $\hat{f}_1$ e $\hat{f}_2$ stimati sul training set: il primo modello è un modello lineare $\hat{f}_1$ (funzione `lm`), mentre il secondo modello è un albero decisionale (funzione `rpart` della libreria `rpart`), entrambi basati su tutte le variabili e con argomenti di *default*. Si applichi l'algoritmo *stacked generalization* e si calcoli l'errore di previsione (RMSE) sul test set, confrontandolo con l'errore di previsione dei singoli modelli

```
rm(list=ls())
library(MASS)
set.seed(123)
istrain = rbinom(n=nrow(Boston),size=1,prob=0.5)>0
train <- Boston[istrain,]
n=nrow(train)
test = Boston[!istrain,-14]
test.y = Boston[!istrain,14]
m=nrow(test)
```

Algoritmo *model stacking*

1. Partition the training data into K folds $\mathcal{F}_1, \ldots, \mathcal{F}_K$
2. For each test fold $\mathcal{F}_k$, $k = 1, \ldots, K$, combine the other $K - 1$ folds to be used as a training fold. For $l = 1, \ldots, L$, fit the lth model to the training fold and make predictions on the test fold $\mathcal{F}_k$. Store these predictions

$$z_i = (\hat{f}_1^{-\mathcal{F}_k}(x_i), \ldots, \hat{f}_L^{-\mathcal{F}_k}(x_i)), \quad i \in \mathcal{F}_k$$

3. Fit the stacking model $\hat{f}_{\text{stack}}$ using

$$(y_1, z_1), \ldots, (y_n, z_n)$$

4. For $l = 1, \ldots, L$, fit the lth model to the full training data and make predictions on the test data. Store these predictions

$$z_i^* = (\hat{f}_1(x_i^*), \ldots, \hat{f}_L(x_i^*)), \quad i = 1, \ldots, m$$

5. Make final predictions $\hat{y}_i^* = \hat{f}_{\text{stack}}(z_i^*)$, $i = 1, \ldots, m$

Esercizio 2

Utilizzare il pacchetto `stacks` per applicare l'algoritmo *model stacking* per il data set `Boston`, con la suddivisione in training e test set specificata nell'esercizio precedente.

Si considerino almeno tre tipologie di modelli, e si calcoli il RMSE sul test set del modello combinato e dei singoli modelli.