Like a Variance

Paola Berchialla, Luca Grassano, Corrado Lanera, Davide Menardi

08 settembre 2015

Analisi

Pacchetti

Per condurre l'analisi abbiamo deciso di utilizzare il software statistico R. Innanzitutto carichiamo quindi i pacchetti che ci saranno utili.

```
library(dplyr)                          # gestione ottimizzata dei data_frame
library(SuperLearner)                                    # Predittore
library(parallel)                 # Abilitazione parallelizzazioni dei processi
```

Import ed esplorazione dei dati

Dopo aver scericato i dati come da istruzioni carichiamoli nel nostro workspace e cominciamo a esplorarli.

```
dat <- as_data_frame(read.csv("dataset_vino_stima.csv"))
#dat %>% glimpse
#summary(dat)
validation <- as_data_frame(read.csv("dataset_vino_previsione.csv"))
#validation %>% glimpse
#summary(validation)
#summary(filter(validation, is.na(Titolo_Alcolometrico_Effettivo)))
```

I dati hanno caratteristiche divisibili in due categorie principali: chimiche e non chimiche.

Dall'osservazione dei dati vediamo che compaiono numerosi NA, sia nell'insieme noto che in quello da etichettare. Possiamo dividere le variabili in NA-variabili (quelle che presentano NA) e in NonNA-variabili (che non presentano alcun NA). Inoltre osserviamo come gli NA siano quasi sempre in egual numero nelle NA-variabili. Da un'esplorazione si evidenzia come un record che presenti una delle NA-variabili con NA, presenta anche tutte le altre NA-variabili con NA.

Non potendo quindi usare NA-varibili per predirre record che presentano NA nelle NA-variabili, decidiamo di considerare solo le NonNA-variabili per la nostra analisi. Osserviamo che le NonNA-variabili sono tutte fattori tranne una (**Qta-Effettiva**). Trasformiamo in un fattore anche quella.

Visto che i nostri algoritmi prevedono una variabile di outcome necessariamente "numeric vector", trasformiamo la variabile **Risposta** come dicotomica 0-1 numerica.

Infine, poiché il contenuto intrinseco delle variabili per noi è del tutto irrilevante, trasformiamo ogni variabile nella codifica numerica che rappresenta l'identificativo sequenziale (1,2,3,...) del suo valore nella sequenza dei valori che assume.

Riformiamo quindi la nuova base di dati come appena descritto, e facciamo lo stesso anche per il dataset su cui operare la previsione.

```
new.dat <- dat %>%
       select(Risposta,
              Data,
              Articolo,
              Tipologia,
              Annata,
              Denominazione,
              Zona_Geografica,
              Tipo,
              Sfuso,
              Millesimato,
              Tirato_Imbottigliato,
              Qta_Effettiva) %>%
       mutate(Qta_Effettiva = as.factor(Qta_Effettiva)) %>%
       mutate(Risposta = as.numeric(as.character(factor(Risposta,
                                                        levels=c('IDONEO',
                                                                 'RIVEDIBILE'
                                                                 ),
                                                        labels=c(0,1)
                                                        )
                                                 )
                                    )
              ) %>%
       mutate_each(funs = "as.numeric")
#glimpse(new.dat)

train <- new.dat
#glimpse(train)
#summary(train)


test <- validation %>%
        select(which(names(validation) %in% names(train))) %>%
        mutate(Qta_Effettiva = as.factor(Qta_Effettiva)) %>%
        mutate_each(funs = "as.numeric")
#glimpse(test)
```

Siamo pronti ad addestrare la nostra tecnica.

Super Learner

La tecnica che adottiamo è la *Super Learner*[1]

Innanzitutto dichiariamo i parametri di risposta e quelli da usare per l'addestramento.

Anche se la variabile di outcome sarà di fatto un valore continuo $[0, 1]$, per noi sarà un risultato dicotomico $0 - 1$, quindi scegliamo per il calcolo dell'errore l'opzione "binomial" per la nostra tecnica. Inoltre come metodo di calcolo per stimare i pesi del predittore utiliziamo quello standard NNLS:

$$\Psi_{SL}(W) = \sum_{j=1}^{k} \alpha_j \Psi_j(W), \qquad \alpha_j \geq 0, \quad \sum \alpha_j = 1;$$

in cui k è il numero dei modelli Ψ_j con $j \in \{1...k\}$.

Dopodiché addestriamo la nostra macchina e valutiamone l'efficacia cross-validando il risultato.

Quindi applichiamola al nostro dataset da analizzare e troviamone le previsioni.

Infine rendiamo il risultato della previsione dicotomico trasformandola in una variabile 0-1 a seconda di un *cut-off* di 0.5.

```
y = "Risposta"
x = setdiff(names(train), y)

family = "binomial"

## Modelli ensembled considerati
SL.library = c('SL.randomForest', 'SL.rpart' , 'SL.glmnet')

## Creazione modello
fit <-  SuperLearner(X = train[,c(x)], Y = as.data.frame(train)[,c(y)],
                     family = 'binomial',
                     SL.library = SL.library,
                     cvControl = list(V = 10, shuffle = TRUE))

## Valutazione performance
pred <-  CV.SuperLearner(X = train[,c(x)], Y = as.data.frame(train)[,c(y)],
                      family = family,
                      SL.library = SL.library,
                      V=10,
                      cvControl = list(V = 5, shuffle = TRUE),
                      parallel = "multicore")
summary(pred)

test <- predict(fit, test)

out <- ifelse(test$pred>=0.5, 1, 0)
```

Esportazione del risultato

Salviamo quindi il risultato come richiesto: in un file *.txt* composto da un'unica colonna.

```
write(out, file = "Like-a-Variance-predict.txt", sep="\n")
```

[1]: M. Laan, E. Polley and E. Hubbart, "Super Learner", Stat. Appl. Genet. Mol. Biol. 2007; 6: Article25.