Aprendizado de Máquinas
Florestas Aleatórias

Douglas Rodrigues

Universidade Federal Fluminense

Alterando hiperparâmetros em `randomForest()`

- Dentre as diversas opções, destacamos duas:
  1. **mtry** = número de variáveis a serem sorteadas.
  2. **ntree** = número de arvores da floresta.

Alterando hiperparâmetros em `randomForest()`

- Dentre as diversas opções, destacamos duas:
  1. **mtry** = número de variáveis a serem sorteadas.
  2. **ntree** = número de arvores da floresta.

```
modelFit<-randomForest(hd ~ ., data=training, ntree=1000,
                       mtry=5, proximity=T)
```

Floresta Aleatória com `train()`

- Iniciamos separando as amostres treino/teste (Validação).

```
> set.seed(100)
> inTrain <- createDataPartition(y=data$hd, p=0.75,list=F)
> training <- data[inTrain,]
> testing <- data[-inTrain,]
```

- Para utilizar floresta aleatória com `train()`, devemos ajustar o método de reamostragem para `"oob"`.

- O padrão do `train()` é construir 500 árvores na floresta. Para alterar, utilizamos a opção `ntree`.

```
> ctrl <- trainControl(method="oob")
> modFit<- train(hd ~ ., data=training,method="rf",
      ntree=50,trControl=ctrl)
```

Floresta Aleatória com `train()`

- O número de variáveis utilizadas para construir cada árvore não é fixo: ele utiliza alguns valores (`mtry`), e seleciona o de melhor acurácia.
- Para fixar o número de variáveis a serem sorteadas, utilizamos o seguinte comando.

```
> ctrl <- trainControl(method="oob")
> tng <- expand.grid(.mtry=7)
> modFit<- train(hd ~ ., data=training,method="rf",
      ntree=50,trControl=ctrl, tuneGrid=tng)
```

Ideia Principal:

- Criar uma sequência de classificadores "fracos" $h_1, h_2, \cdots, h_k$.
- Baseado nas taxas de erro dos estimadores, considerar pesos $\alpha_1, \alpha_2, \cdots, \alpha_k$.
- Criar um classificador final que combina os classificadores fracos, considerando os pesos.