Lesson 10 Classifier combination methods

Lusine Zilfimian

May 11 (Monday), 2020

Contents

- Quiz

Contents

- Quiz
- Intro to EM

Contents

- Quiz
- Intro to EM
- Bagging, Boosting

Contents

Last Lecture ReCap

- What are the differences between K-means and HC?

Last Lecture ReCap

- What are the differences between K-means and HC?
- Which type of Proximity between Clusters do you know?

Last Lecture ReCap

- What are the differences between K-means and HC?
- Which type of Proximity between Clusters do you know?
- Disadvantages of k-means

EM

Suggested materials to read (master) regression

- G. James, D. Witten, et al., An Introduction to Statistical Learning, Chapter 8.2, 8.3

EM

Suggested materials to read (master) regression

- G. James, D. Witten, et al., An Introduction to Statistical Learning, Chapter 8.2, 8.3
- Tan P., et al., Introduction to Data Mining, Chapter 5.6

EM

Suggested materials to read (master) regression

- G. James, D. Witten, et al., An Introduction to Statistical Learning, Chapter 8.2, 8.3
- Tan P., et al., Introduction to Data Mining, Chapter 5.6
- Articles of Breiman and Schapire

Idea of EM

- An ensemble method constructs a set of **base classifiers** from training data and performs classification by taking a **vote** on the predictions made by each base classifier

Two necessary conditions

Idea of EM

- An ensemble method constructs a set of **base classifiers** from training data and performs classification by taking a **vote** on the predictions made by each base classifier

Two necessary conditions

- the base classifiers should be independent of each other

Idea of EM

- An ensemble method constructs a set of **base classifiers** from training data and performs classification by taking a **vote** on the predictions made by each base classifier

Two necessary conditions

- the base classifiers should be independent of each other
- the base classifiers should do better than a classifier that performs random guessing

Methods for Constructing an Ensemble Classifier

- By manipulating the training set (Bagging, Boosting)

Methods for Constructing an Ensemble Classifier

- By manipulating the training set (Bagging, Boosting)
- By manipulating the input features (Random Forest)

Methods for Constructing an Ensemble Classifier

- By manipulating the training set (Bagging, Boosting)
- By manipulating the input features (Random Forest)
- By manipulating the class labels

Methods for Constructing an Ensemble Classifier

- By manipulating the training set (Bagging, Boosting)
- By manipulating the input features (Random Forest)
- By manipulating the class labels
- By manipulating the learning algorithm

Training set

- The size of the training set is often kept the same as the original data

Training set

- The size of the training set is often kept the same as the original data
- The variability among training examples is one of the primary sources of errors in a classifier

Training set

- The size of the training set is often kept the same as the original data
- The variability among training examples is one of the primary sources of errors in a classifier
- Base classifiers built from different training sets may help to reduce such types of errors

Bootstrap

Bootstrap

- Repeatedly sampling observations from the original data set

Bootstrap

Bootstrap

- Repeatedly sampling observations from the original data set
- The sampling is performed with replacement

Bootstrap

```r
library(boot) # for Bootstrap
library(ISLR) # for Portfolio dataset
pt <- Portfolio
str(pt)
```

```
## 'data.frame':    100 obs. of  2 variables:
##  $ X: num  -0.895 -1.562 -0.417 1.044 -0.316 ...
##  $ Y: num  -0.235 -0.885 0.272 -0.734 0.842 ...
```

This data is used to estimate the optimal fraction to invest in each asset to minimize investment risk of the combined portfolio.

Bootstrap

```
alpha.hat = function(index){
  X = pt$X[index]
  Y = pt$Y[index]
  return((var(Y) - cov(X,Y))/( var(X) + var(Y) - 2 * cov(X,Y))
```

Index - vector indicating which observations should be used to estimate apha. All observations:

```
alpha.hat(1:100)
```

```
## [1] 0.5758321
```

Resampling

To randomly select 100 observations from the range 1 to 100, with replacement we will use the function sample()

```
set.seed(2708)
(a1 <- alpha.hat(sample(x = 1:100, size = 100, replace = T)))
```

```
## [1] 0.6232892
```

```
set.seed(1105)
(a2 <- alpha.hat(sample(x = 1:100, size = 100, replace = T)))
```

```
## [1] 0.5660552
```

```
set.seed(1977)
(a3 <- alpha.hat(sample(x = 1:100, size = 100, replace = T)))
```

```
## [1] 0.6571697
```

Bootstrap

- boot() function automates this approach, R number of bootstrap replicates

```
boot(data = pt, statistic =  alpha.hat, R = 3)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = pt, statistic = alpha.hat, R = 3)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.5758321 -0.01375633  0.06027934
```

Bagging is bootstrap aggregating

- We generally do not have access to multiple training sets

Bagging is bootstrap aggregating

- We generally do not have access to multiple training sets
- Repeatedly samples with replacement

Bagging is bootstrap aggregating

- We generally do not have access to multiple training sets
- Repeatedly samples with replacement
- Uses origina data set

Bagging is bootstrap aggregating

- We generally do not have access to multiple training sets
- Repeatedly samples with replacement
- Uses origina data set
- Uses uniform probability distribution

Bagging is bootstrap aggregating

- We generally do not have access to multiple training sets
- Repeatedly samples with replacement
- Uses origina data set
- Uses uniform probability distribution
- Size of each bootstrap sample = original data Size

Bagging

- Trains k classifiers

Bagging

- Trains k classifiers
- Uses majority of voting

Bagging

- Trains k classifiers
- Uses majority of voting
- Example using **Decision Stump**

Bagging

- Trains k classifiers
- Uses majority of voting
- Example using **Decision Stump**
- Reduces the variance of a statistical learning method

Boosting

- Boosting works in a similar way, except that the classifiers are created sequentially.

Boosting

- Boosting works in a similar way, except that the classifiers are created sequentially.
- Boosting is iterative procedure

Boosting

- Boosting works in a similar way, except that the classifiers are created sequentially.
- Boosting is iterative procedure
- Change the train set such that base classifiers will focus on examples that are hard to classify

Boosting

- Initially the examples are assigned equal weights

Boosting

- Initially the examples are assigned equal weights
- Classifier is induced from the training set

Boosting

- Initially the examples are assigned equal weights
- Classifier is induced from the training set
- Classify all the examples in the original data

Boosting

- Initially the examples are assigned equal weights
- Classifier is induced from the training set
- Classify all the examples in the original data
- The weights of the training examples are updated at the end of each boosting round

Boosting

- Initially the examples are assigned equal weights
- Classifier is induced from the training set
- Classify all the examples in the original data
- The weights of the training examples are updated at the end of each boosting round
- Examples that were not chosen in the previous round have a better chance of being selected in the next round since their predictions in the previous round were likely to be wrong.

Random Forest

- Ensemble method for decision tree

Random Forest

- Ensemble method for decision tree
- Bagging using decision trees is a special case of random forest

Random Forest

- Ensemble method for decision tree
- Bagging using decision trees is a special case of random forest
- Trees become more correlated thus the generalization error bound tends to increase

Random forests

- We build a number of decision trees on bootstrapped training samples

Random forests

- We build a number of decision trees on bootstrapped training samples
- Each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.

Random forests

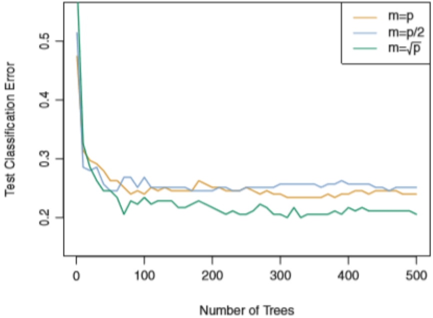
- We build a number of decision trees on bootstrapped training samples
- Each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.
- This may help reduce the bias present in the resulting tree

Random forests

- We build a number of decision trees on bootstrapped training samples
- Each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.
- This may help reduce the bias present in the resulting tree
- Since only a subset of the features needs to be examined at each node, this approach helps to significantly reduce the runtime of the algorithm.

Random forests

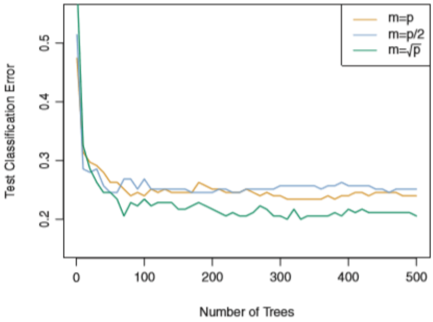
- Problem with small number of p



Source: G. James, D. Witten, et al., *An Introduction to Statistical Learning*

Random forests

- Problem with small number of p
- Choosing m



Source: G. James, D. Witten, et al., *An Introduction to Statistical Learning*

Ideas for Project

- AdaBoost

Ideas for Project

- AdaBoost
- Comparison of all model (both theoretically and empirically)