Predicting
June 21 Tailin Wu
Outline

• Review: decision trees
  • Summary
  • Measures of Error
  • Bias and variance

• Reduce variance: combining predictors
  • Bootstrapping
  • Random forests
  • Boosting
  • Combining different classifiers

• Time series and prediction
Outline

• Review: decision trees
  • Summary
  • Measures of Error
  • Bias and variance

• Reduce variance: combining predictors
  • Bootstrapping
  • Random forests
  • Boosting
  • Combining different classifiers

• Time series and prediction
Summary: Decision Trees
Summary: Decision Trees

Measures of impurity

\[ p_{mk} = \sum_{x_i \text{ in leaf } m} 1 \left( x_i = k \right) \]

<table>
<thead>
<tr>
<th>Measures of impurity</th>
<th>Formula</th>
<th>Perfect purity</th>
<th>No purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misclassification Error</td>
<td>( 1 - p_{mk(m)}, k(m): \text{the most common } k )</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Gini index</td>
<td>( 1 - \sum_{k=1}^{K} p_{mk}^2 )</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Information gain</td>
<td>( - \sum_{k=1}^{K} p_{mk} \log_2 p_{mk} )</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Reference: [Decision Trees Learning wiki](http://Decision Trees Learning wiki)
Bias and variance

Goal: estimate $Y = f(X) + \epsilon$ with $\hat{f}(X)$

$$Err(x) = E[(Y - \hat{f}(x))^2]$$

$$= (E[f(x)] - \hat{f}(x))^2$$  \hspace{1cm} \text{Bias}^2

$$+ Var[\hat{f}(x)]$$  \hspace{1cm} \text{Variance}

$$+ E[\epsilon^2]$$  \hspace{1cm} \text{Irreducible Error}

Decision Trees: \textbf{low bias, high variance}

To decrease variance: bootstrapping
Outline

• Review: decision trees
  • Summary
  • Measures of Error
  • Bias and variance

• Reduce variance: combining predictors
  • Bootstrapping
  • Random forests
  • Boosting
  • Combining different classifiers

• Time series and prediction
Bagging

Bagging: Bootstrap Aggregating

library(ElemStatLearn); data=(ozone, "ElemStatLearn")
ozone <- ozone[order(ozone$ozone),]
head(ozone)
ll <- matrix(NA, nrow=10, ncol=155)
for(i in 1:10){
  ss <- sample(1:dim(ozone)[1], replace=T)
ozone0 <- ozone[ss, ]; ozone0 <- ozone0[order(ozone0$ozone),]
  loess0 <- loess(temperature ~ ozone, data=ozone0, span=0.5)
  ll[i,] <- predict(loess0, newdata=data.frame(ozone=1:155))
}
plot(ozone$ozone, ozone$temperature, pch=19, cex=0.5)
for(i in 1:10){lines(1:155, ll[i,], col="grey", lwd=2)}
lines(1:155, apply(ll, 2, mean), col="red", lwd=2)
Bagging

Bagging: reduce variance

Ellipsoid separation →
Two categories,
Two predictors

Single tree decision boundary
100 bagged trees..
Random forest

- Bootstrap samples
- Bootstrap features at each node

The ensemble model

Forest output probability $p(c|v) = \frac{1}{T} \sum_{t=1}^{T} p_{t}(c|v)$
Random forest: algorithm

Begin

For each tree

Chose training data subset

Stop condition holds at each node?

Yes

No

Build the next split

Calculate prediction error (4)

End

Chose variable subset

For each chosen variable

Sample data (1)

Sort by the variable (2)

Compute Gini index at each split point (3)

Chose the best split
Random forest: features & advantages

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases. \( \log_2(N) \)
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
Random forest: features & advantages

• It has methods for balancing error in class population unbalanced data sets.

• Generated forests can be saved for future use on other data.

• Prototypes are computed that give information about the relation between the variables and the classification.

• It computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data.

• The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.

• It offers an experimental method for detecting variable interactions.
Boosting

1. Take lots of (possibly) weak predictors
2. Weight them and add them up
3. Get a strong predictor

Round 1:

\[ h_1 \]
\[ \epsilon_1 = 0.30 \]
\[ \alpha_1 = 0.42 \]
Boosting

Goal: minimize error

Iterate:

1. Set a classifier $h$
2. Calculate weights based on errors
3. Upweight missed classifications and select next $h$
Boosting

Goal: minimize error

Iterate:

1. Set a classifier $h$
2. Calculate weights based on errors
3. Upweight missed classifications and select next $h$
Combining predictors: a general approach

Netflix Prize ($1M):

BellKor: Combination of 107 predictors
Combining predictors: a general approach

Heritage Health Prize ($3M):

- Predict which people will go back to the hospital based on their health record
- Won by teams that applies emsembling to different predictors
Combining predictors

Basic Intuition: Majority votes

Suppose we have 5 completely independent classifiers

If accuracy is 70% for each:

- \( 10 \times (0.7)^3 (0.3)^2 + 5 \times (0.7)^4 (0.3)^2 + (0.7)^5 \)
- 83.7% majority vote accuracy

With 101 independent classifiers

- 99.9% majority vote accuracy
Combining predictors

Combining similar classifiers:
• Bagging, bootstrapping, random forests

Combining different classifiers:
• Model stacking
• Model ensembling
Combining predictors

Model stacking:

Training a learning algorithm to combine the predictions of several other learning algorithms

Steps:

1. All of the other algorithms are trained using the available data
2. A combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs

Model ensembling
Outline

• Review: decision trees
  • Summary
  • Measures of Error
  • Bias and variance

• Reduce variance: combining predictors
  • Bootstrapping
  • Random forests
  • Boosting
  • Combining different classifiers

• Time series and prediction
Time series and forecasting

Time series examples:
- Temperature
- Stock price
- PM2.5
- Epidemic spread

Netflix, Inc.
NASDAQ: NFLX - Jun 19 4:00 PM EDT

657.10 ↓ 6.10 (0.92%)

1 day  5 day  1 month  3 month  1 year  5 year  max

Mar 27  Apr 10  Apr 24  May 8  May 22  Jun 5  Jun 19

Open 673.70  Market cap 39.64B
High 674.93  P/E ratio (ttm) 170.98
Low 656.75  Dividend yield -
Time series and forecasting

What is different:

- Data are dependent over time
- Specific pattern types
  - Trends - long term increase or decrease
  - Seasonal patterns - patterns related to time of week, month, year, etc.
  - Cycles - patterns that rise and fall periodically
- Subsampling into training/test is more complicated
- Similar issues arise in spatial data
  - Dependency between nearby observations
  - Location specific effects
- Typically goal is to predict one or more observations into the future.
- All standard predictions can be used (with caution!)
Time series and forecasting

Things to beware of:

1. Spurious correlations

http://www.tylervigen.com/spurious-correlations

Also common: geographical data
Time series and forecasting

Things to beware of:

2. Extropolation
Time series and forecasting

Techniques

Moving average:

\[ Y_t = \frac{1}{2 \cdot k + 1} \sum_{j=-k}^{k} y_{t+j} \]

Exponential smoothing:

\[ \hat{y}_{t+1} = \alpha y_t + (1 - \alpha) \hat{y}_{t-1} \]
Summary

• Review: decision trees
  • Summary
  • Measures of Error
  • Bias and variance

• Reduce variance: combining predictors
  • Bootstrapping
  • Random forests
  • Boosting
  • Combining different classifiers

• Time series and prediction
Reference

- Courera – Practical Machine learning
- Wikipedia – Decision tree learning
- Wikipedia – Model learning
- Random forecasts – ppt by Albert Montillo
- Random forests – ppt by Predrag