Advanced R Programming - Lecture 7

Krzysztof Bartoszek
(slides by Leif Jonsson and Måns Magnusson)

Linköping University

krzysztof.bartoszek@liu.se

29 September 2017

Today

Data munging

Machine Learning

Supervised learning in R

Probability in R

Big data

Questions since last time?

Tidy data

Tidy data and **messy** data

Tidy data

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

Tidy data

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

Examples: `iris` and `faithful`

Why tidy?

80 % of Big Data work is data munging

Why tidy?

80 % of Big Data work is data munging

Analysis and visualization is based on tidy data

Why tidy?

80 % of Big Data work is data munging

Analysis and visualization is based on tidy data

Performant code

Data analysis pipeline

Messy data $\rightarrow$ Tidy data $\rightarrow$ Analysis

Messy data

1. Column headers are values, not variable names. (`AirPassengers`)

Messy data

1. Column headers are values, not variable names. (`AirPassengers`)
2. Multiple variables are stored in one column. (`mtcars`)

Messy data

1. Column headers are values, not variable names. (`AirPassengers`)
2. Multiple variables are stored in one column. (`mtcars`)
3. Variables are stored in both rows and columns. (`crimtab`)

Messy data

1. Column headers are values, not variable names. (`AirPassengers`)
2. Multiple variables are stored in one column. (`mtcars`)
3. Variables are stored in both rows and columns. (`crimetab`)
4. Multiple types of observational units are stored in the same table.

Messy data

1. Column headers are values, not variable names. (`AirPassengers`)
2. Multiple variables are stored in one column. (`mtcars`)
3. Variables are stored in both rows and columns. (`crimetab`)
4. Multiple types of observational units are stored in the same table.
5. A single observational unit is stored in multiple tables.

dplyr

Verbs for handling data

Highly optimized C++ code (backend)

Handling larger datasets in R
(no copy-on-modify)

dplyr+tidyr

https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf
the cheatsheet

Regular Expressions

Language for manipulating strings

Find strings that match a pattern

Extract patterns from strings

Replace patterns in strings

Component in many functions
(grep, gsub, stringr::, tidyr::)

Regular Expressions - Syntax

fruit <- c("apple", "banana", "pear", "pineapple")

Symbol	Description	Example
?	The preceding item is optional and will be matched at most once	grep("pi?",fruit)
*	The preceding item will be matched zero or more times	grep("pi*",fruit)
+	The preceding item will be matched one or more times	grep("pi+",fruit)
n	The preceding item is matched exactly n times	grep("p{2}",fruit)

Regex Examples - Finding matching

```
> library(gapminder)
> grep("we", gapminder$country)
[1] 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475
1695 1696 1697
[18] 1698 1699 1700 1701 1702 1703 1704
grep("we", gapminder$country, value=TRUE)
[1] "Sweden"   "Sweden"   "Sweden"   "Sweden"   "Sweden"
"Sweden"  "Sweden"   "Sweden"
[9] "Sweden"   "Sweden"   "Sweden"   "Sweden"   "Zimbabwe"
"Zimbabwe" "Zimbabwe"
[17] "Zimbabwe" "Zimbabwe" "Zimbabwe" "Zimbabwe" "Zimbabwe"
"Zimbabwe" "Zimbabwe"
```

Regex Examples - Extraction

```
> library(stringr)
> strs <- c("The 13 Cats in the Hats are 17 years", "4 score
ago")
> str_extract_all(strs, "[a-z]+")
[[1]]
[1] "he"    "ats"   "in"    "the"   "ats"   "are"   "years"
[[2]]
[1] "scores" "and"    "beers"  "ago"
> str_extract(strs, "[a-z]+")
[1] "he"     "scores"
> str_extract(strs, "[0-9]+")
[1] "13" "4"
> str_extract_all(strs, "[0-9]+")
[[1]]
[1] "13" "17"
[[2]]
[1] "4" "7"
```

Regex Examples - Tidyr separate

```
> library(gapminder)
> # Create artificial column with numeric data in text
> rnds <- ceiling(runif(nrow(gapminder),80,200))
> gapminder$country <- paste(gapminder$country, rnds, " population")
> tdy <- gapminder %>% separate(country, into = c("Count", "CPop"), sep
="\\d+")
> head(tdy)
# A tibble: 6 <U+00D7> 7
Count             CPop          continent  year lifeExp      pop gdpPercap
<chr>             <chr>          <fctr> <int>   <dbl>    <int>     <dbl>
1 Afghanistan     population       Asia  1952  28.801  8425333  779.4453
2 Afghanistan     population       Asia  1957  30.332  9240934  820.8530
3 Afghanistan     population       Asia  1962  31.997 10267083  853.1007
4 Afghanistan     population       Asia  1967  34.020 11537966  836.1971
```

Machine learning?

Automatically detect patterns in data

Machine learning?

Automatically detect patterns in data

Predict future observation

Machine learning?

Automatically detect patterns in data

Predict future observation

Decision making under uncertainty

Types of Machine learning

Supervised learning

Types of Machine learning

Supervised learning

Unsupervised learning

Types of Machine learning

Supervised learning

Unsupervised learning

Reinforcement learning

Supervised learning

(also called predictive learning)

response variable

covariates/features

training set

$$D = (x_i, y_i)_{(i=1)}^{N}$$

Supervised learning examples

If y_i is categorical:
classification

If y_i is real:
regression

Unsupervised learning

(also called knowledge discovery)

dimensionality reduction

latent variable modeling

$$D = (x_i)_{(i=1)}^{N}$$

clustering, PCA, discovering of graph structures

data visualization

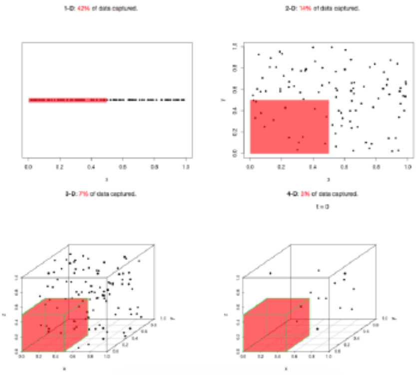
Curse of dimensionality

The more variables the larger distance between datapoints

Euclidian metric

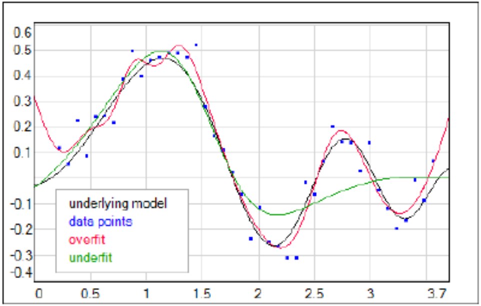
$$d^2(\vec{x},\vec{y}) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2 + \ldots$$

Curse of dimensionality



http://www.newsnshit.com/curse-of-dimensionality-interactive-demo/

Fit (bias) and variance in ML



Underfit = bad fit, low variance
Overfit = good fit, high variance

NetMaker (Neural networks simulator and designer by Robert Sulej, Warsaw Univ. Tech.):

`http://www.ire.pw.edu.pl/~rsulej/NetMaker/index.php?pg=e06`

Model selection

fit and variance - tradeoff

Model selection

fit and variance - tradeoff

hyper parameters

Model selection

fit and variance - tradeoff

hyper parameters

generalization error

Model selection

fit and variance - tradeoff

hyper parameters

generalization error

validation set/cross validation

Model selection

fit and variance - tradeoff

hyper parameters

generalization error

validation set/cross validation

information criteria: model fit penalized for model dimensionality

Predictive modeling pipeline

1. Set aside data for test (estimate generalization error)
2. Set aside data for validation (if hyperparams)
3. Run algorithms
4. Find best/optimal hyperparameters (on validation set)
5. Choose final model
6. Estimate generalization error on test set

No free lunch

different models work in different domains

No free lunch

different models work in different domains

accuracy-complexity-intepretability tradeoff

No free lunch theorem

different models work in different domains

accuracy-complexity-intepretability tradeoff

...but more data always wins

the caret package

package for supervised learning

the caret package

package for supervised learning

does not contain methods - a framework

the caret package

package for supervised learning

does not contain methods - a framework

compare methods on hold-out-data

the caret package

package for supervised learning

does not contain methods - a framework

compare methods on hold-out-data

`http://topepo.github.io/caret/`

the caret package

package for supervised learning

does not contain methods - a framework

compare methods on hold-out-data

`http://topepo.github.io/caret/`

specific algorithms are part of other courses

Probability Functions

Prefix	Description	Example
r	Random draw	rnorm
d	Density function	dbinom
q	Quantile function	qbeta
p	CDF	pgamma

Big data

Today's trend:

- whole genome
- surveillance cameras (CCTV)
- Internet traffic
- credit card transactions
- everything communicating with everything

Big data is relative...

... to computational complexity

$$O(N) \quad 10^{12}$$

Big data is relative...

... to computational complexity

$$
\begin{array}{ll}
O(N) & 10^{12} \\
O(N^2) & 10^{6}
\end{array}
$$

Big data is relative...

... to computational complexity

$O(N)$	10^{12}
$O(N^2)$	10^{6}
$O(N^3)$	10^{4}

Big data is relative...

... to computational complexity

$O(N)$	10^{12}
$O(N^2)$	10^6
$O(N^3)$	10^4
$O(2^N)$	50

Big data is relative...

... to computational complexity

$O(N)$	10^{12}
$O(N^2)$	10^6
$O(N^3)$	10^4
$O(2^N)$	50

We need algorithms that scale!

Big data is relative...

... to computational complexity

$O(P^2 * N)$	Linear regression
$O(N^3)$	Gaussian processes
$O(N^2)/O(N^3)$	Support vector machines
$O(T(P * N * log(N))$	Random forests
$O(I * N)$	Topic models

Big data in R

R stores data in RAM

Big data in R

R stores data in RAM

integers
4 bytes
numerics
8 bytes

Big data in R

R stores data in RAM

integers
4 bytes
numerics
8 bytes

A matrix with 100M rows and 5 cols with numerics
$$100000000 * 5 * 8/(1024^3) \approx 3.8GB$$

`help(Memory)`; `help("Memory-limits")`
Genome storage ... ?

How to deal with large data sets

Handle chunkwise
Subsampling
More hardware
C++/Java backend (dplyr)
Reduce data in memory
Database backend

If not enough

Spark and SparkR

Fast cluster computations for ML /STATS

Introduction to Spark:
https://www.youtube.com/watch?v=_Ss1Cm6WO-I

The End... for today.

Questions?

See you next time!