Introduction to Machine Learning
NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká **Martin Holub**

{Hladka | Holub}@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Programming questions

- **(Hierarchichal) clustering**
  - Feature scaling
  - `NLI` data set (75 documents, 5 languages)
- **Gradient descent algorithm**
  - Find a minimum of a function using Gradient Descent Algorithm (simple illustration)
- **`Auto` data set**
  - Compute Pearson's correlation coeffcients for `mpg`, `displacement`, `weight`, `horsepower`, `acceleration` in the `Auto` data set
  - Draw boxplots to visualize comparison `mpg` by `origin`, `mpg` by `model year`, and `weight` by `origin`
- **Linear regression**
  - `Auto` data set, target attribute: `mpg`

Feature scaling

Different ranges and units of features

- Is the engine displacement more significant than mpg/cylinders/acceleration?

```
> str(Auto)
'data.frame':   392 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
 $ weight      : num  3504 3693 3436 3433 3449 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : Factor w/ 3 levels "USA","Europe",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231
```

Feature scaling

Scaling

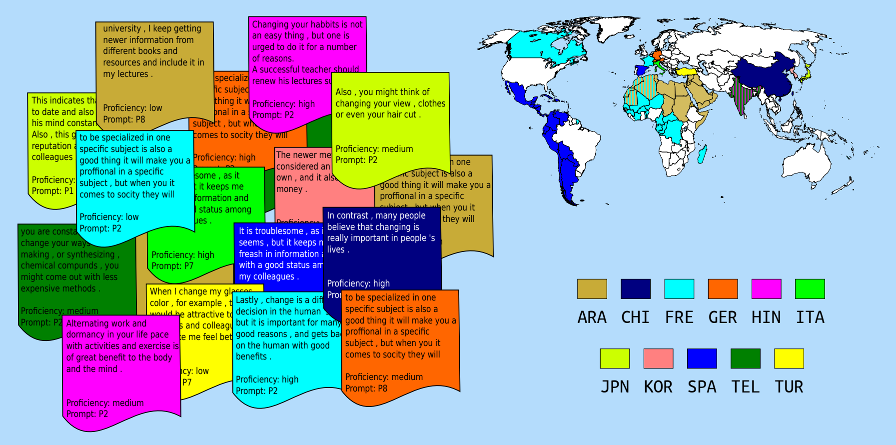
- normalization $z = \frac{x - x_{min}}{x_{max} - x_{min}}$
  $z \in < 0, 1 >$
- standardization $z = \frac{x - \overline{x}}{sd_x}$
  $\overline{z} = 0, sd_z = 1$

Useful especially for

- Gradient Descent Based Algorithms
- Distance based algorithms

```
> head(scale(Auto[,c('mpg', 'displacement', 'weight')]))
         mpg displacement    weight
1 -0.6977467     1.075915 0.6197483
2 -1.0821153     1.486832 0.8422577
3 -0.6977467     1.181033 0.5396921
4 -0.9539925     1.047246 0.5361602
5 -0.8258696     1.028134 0.5549969
6 -1.0821153     2.241772 1.6051468
```

Native language identification task (NLI)



NLI

Identifying the native language (L1) of a writer based on a sample of their writing in a second language (L2)

Our data

- **L1s**: Arabic (ARA), Chinese (ZHO), French(FRA), German (DEU) Hindi (HIN), Italian (ITA), Japanese (JPN), Korean (KOR), Spanish (SPA), Telugu (TEL), Turkish (TUR)
- **L2**: English
- **Real-world objects**: For each L1, 1,000 texts in L2 from The ETS Corpus of Non-Native Written English (former TOEFL11), i.e. $Train \cup DevTest$
- **Target class:** L1

More detailed info is available at the course website.

NLI

Topic
Most advertisements make products seem much better than they really are

Sample text
now a days the publisity is the best way to promoved a produt and if you wanth to sale a product you should bring some information that makes , that the people who is seeing the advertisements make sure that the product very good and in the future this person could buy it .

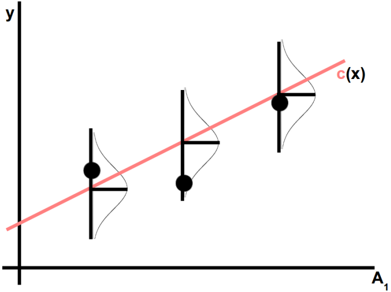
L1 = Spanish

Linear regression
Random error term

- numerical target attribute Y
- $\mathbf{y} = \mathbf{X}\Theta^\top + \epsilon$
- random error term ϵ having mean zero, very often unobserved

Linear regression
Random error term

- $\epsilon_i = y_i - \Theta^\top \mathbf{x}_i$ (true target value y_i, expected value $\Theta^\top \mathbf{x}_i$)
- Assumption like: At each value of A_1, the output value y is subject to random error ϵ that is normally distributed $N(0, \sigma^2)$



Linear regression
Random error term

- $\epsilon_i = y_i - \Theta^\top \mathbf{x}_i$ (true target value y_i, expected value $\Theta^\top \mathbf{x}_i$)
- residual $e_i = y_i - \hat{\Theta}^\top \mathbf{x}_i$ is an estimate of ϵ_i