

# Introduction to Machine Learning

## NPFL 054

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

Barbora Hladká  
hladka@ufal.mff.cuni.cz

Martin Holub  
holub@ufal.mff.cuni.cz

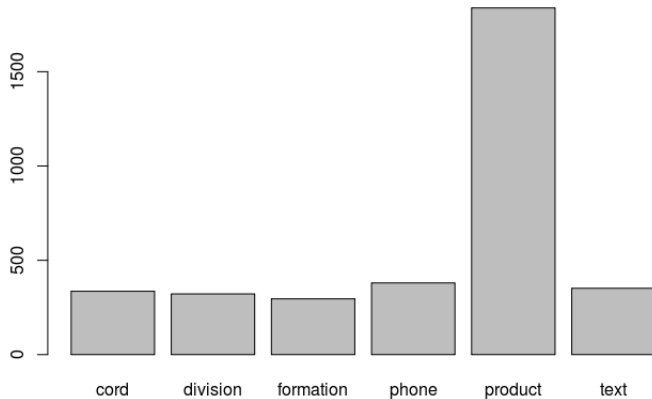
Charles University,  
Faculty of Mathematics and Physics,  
Institute of Formal and Applied Linguistics

## Outline

- **Entropy and conditional entropy**
  - definition, calculation, and meaning
  - application for feature selection

# WSD task — distribution of target class values

```
> examples <- read.table("wsd.development.csv", header=T)
> plot(examples$SENSE)
>
```



# Amount of information contained in a value?

How much information do you gain when you observe a random event?

According to the **Information Theory**, **amount of information** contained in an event is given by

$$I = \log_2 \frac{1}{p} = -\log_2 p$$

where  $p$  is probability of the event occurred.

Thus, the lower probability, the more information you get when you observe an event (e.g. a feature value). If an event is certain ( $p = 100\%$ ), then the amount of information is zero.

# Amount of information in SENSE values

```
### probability distribution of SENSE
> round(table(examples$SENSE)/nrow(examples), 3)

      cord  division formation      phone  product      text
0.095    0.091    0.084    0.108    0.522    0.100
>

### amount of information contained in SENSE values
> round(-log2(table(examples$SENSE)/nrow(examples)), 3)

      cord  division formation      phone  product      text
3.391    3.452    3.574    3.213    0.939    3.324
>
```

**What is the average amount of information that you get when you observe values of the attribute SENSE?**

# Entropy

The average amount of information that you get when you observe random values is

$$\sum_{value} \Pr(value) \cdot \log_2 \frac{1}{\Pr(value)} = - \sum_{value} \Pr(value) \cdot \log_2 \Pr(value)$$

**This is what information theory calls *entropy*.**

- Entropy of a random variable  $X$  is denoted by  $H(X)$ 
  - or,  $H(p_1, p_2, \dots, p_n)$  where  $\sum_{i=1}^n p_i = 1$
- Entropy is a measure of the uncertainty in a random variable
  - or, measure of the uncertainty in a probability distribution
- The unit of entropy is bit; entropy says how many bits *on average* you necessarily need to encode a value of the given random variable

# Properties of entropy

## Normality

$$H\left(\frac{1}{2}, \frac{1}{2}\right) = 1$$

## Continuity

$H(p, 1 - p)$  is a continuous function

## Non negativity and maximality

$$0 \leq H(p_1, p_2, \dots, p_n) \leq H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$$

## Symmetry

$H(p_1, p_2, \dots, p_n)$  is a symmetric function of its arguments

## Recursivity

$$H(p_1, p_2, p_3, \dots, p_n) = H(p_1 + p_2, p_3, \dots, p_n) + (p_1 + p_2)H\left(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2}\right)$$

# Entropy of SENSE

**Entropy of SENSE is 2.107129 bits.**

```
### probability distribution of SENSE
> p.sense <- table(examples$SENSE)/nrow(examples)
>
### entropy of SENSE
> H.sense <- - sum( p.sense * log2(p.sense) )
> H.sense
[1] 2.107129
```

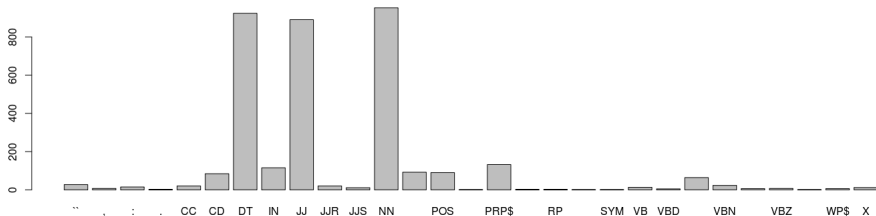
**The maximum entropy value would be  $\log_2(6) = 2.584963$  if and only if the distribution of the 6 senses was uniform.**

```
> p.uniform <- rep(1/6, 6)
> p.uniform
[1] 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
>
### entropy of uniformly distributed 6 senses
> - sum( p.uniform * log2(p.uniform) )
[1] 2.584963
```



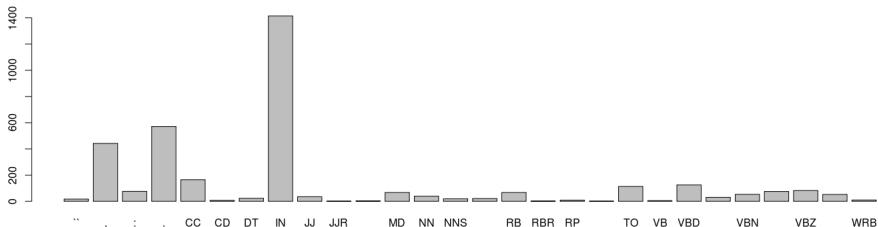
# Distribution of feature values – A16

```
> levels(examples$A16)
[1] "" ",'" ":" ". ." "CC" "CD" "DT" "IN" "JJ"
[10] "JJR" "JJS" "NN" "NNS" "POS" "PRP" "PRP$" "RB" "RP"
[19] "-RRB-" "SYM" "VB" "VBD" "VBG" "VBN" "VBP" "VBZ" "WDT"
[28] "WP$" "X"
> plot(examples$A16)
>
```



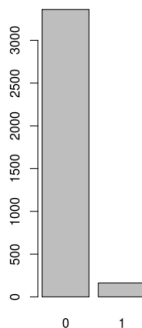
# Distribution of feature values – A17

```
> levels(examples$A17)
[1] "''"      ", "      ":"      "."      "CC"     "CD"     "DT"     "IN"     "JJ"
[10] "JJR"    "-LRB-"  "MD"     "NN"     "NNS"    "PRP"    "RB"     "RBR"    "RP"
[19] "-RRB-"  "TO"     "VB"     "VBD"    "VBG"    "VBN"    "VBP"    "VBZ"    "WDT"
[28] "WRB"
> plot(examples$A17)
>
```



# Distribution of feature values – A4

```
> levels(examples$A4)
[1] "0" "1"
>
```



# Entropy of features

**Entropy of A16 is 2.78 bits.**

```
> p.A16 <- table(examples$A16)/nrow(examples)
> H.A16 <- - sum( p.A16 * log2(p.A16) )
> H.A16
[1] 2.777606
```

**Entropy of A17 is 3.09 bits.**

```
> p.A17 <- table(examples$A17)/nrow(examples)
> H.A17 <- - sum( p.A17 * log2(p.A17) )
> H.A17
[1] 3.093003
```

**Entropy of A4 is 0.27 bits.**

```
> p.A4 <- table(examples$A4)/nrow(examples)
> H.A4 <- - sum( p.A4 * log2(p.A4) )
> H.A4
[1] 0.270267
```

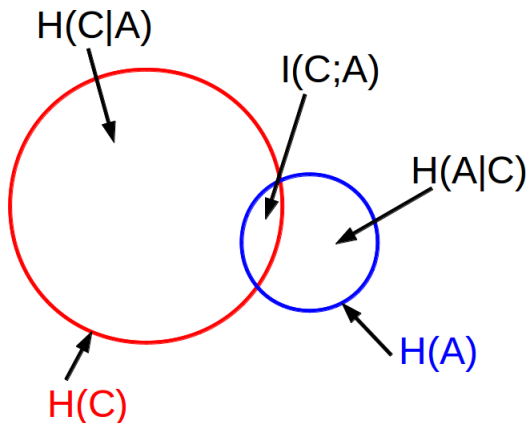
# Conditional entropy $H(C | A)$

How much does target class entropy decrease if we have the knowledge of a feature?

The answer is **conditional entropy**:

$$H(C | A) = - \sum_{y \in C, x \in A} \Pr(y, x) \cdot \log_2 \Pr(y | x)$$

# Conditional entropy and mutual information



## WARNING

There are NO SETS in this picture! Entropy is a quantity, only a number!

# Conditional entropy and mutual information

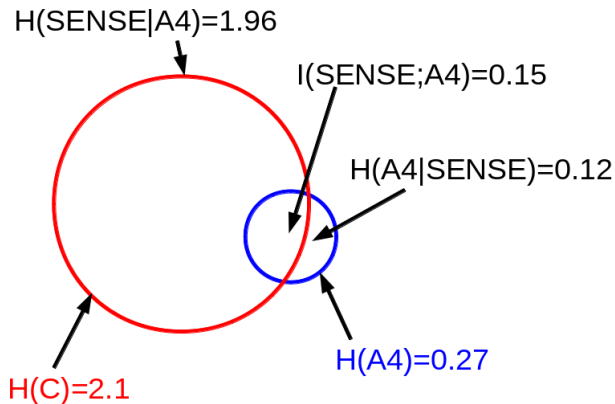
**Mutual information** measures the amount of information that can be obtained about one random variable by observing another.

Mutual information is a symmetrical quantity.

$$H(C) - H(C|A) = I(C;A) = H(A) - H(A|C)$$

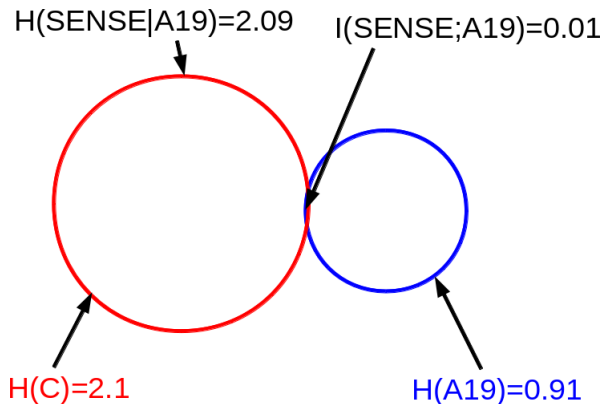
Another name for mutual information is **information gain**.

# Conditional entropy – feature A4

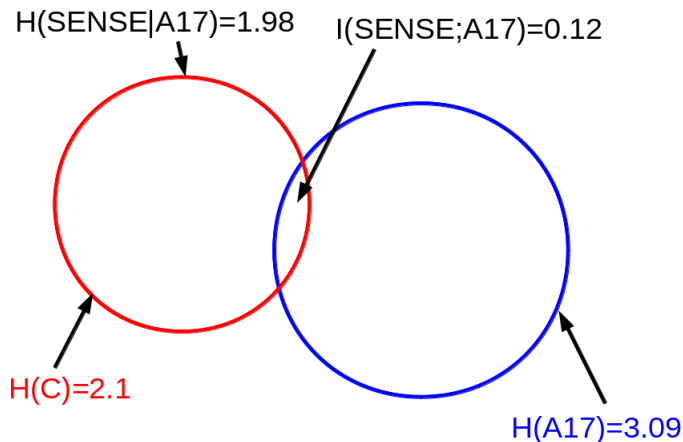




# Conditional entropy – feature A19



# Conditional entropy – feature A17



# User-defined functions in R

## Structure of a user-defined function

```
myfunction <- function(arg1, arg2, ... ){  
  ... statements ...  
  return(object)  
}
```

Objects in a function are local to the function.

## Example – a function to calculate entropy

```
> entropy <- function(x){  
+   p <- table(x) / NROW(x)  
+   return( -sum(p * log2(p)) )  
+ }  
>  
  
# invoking the function  
> entropy(examples$SENSE)  
[1] 2.107129
```

## Summary

- **Information theory provides a measure** for comparing how the knowledge of features *statistically* contribute to the knowledge about target class.
- The lower conditional entropy  $H(C | A)$ , the better chance that  $A$  is a useful feature.
- However, since features typically interact, conditional entropy  $H(C | A)$  should NOT be the only criterion when you do feature selection. You need experiments to see if a feature with high information gain really helps.

## Note

Also, decision tree learning algorithm makes use of entropy when it computes purity of training subsets.

## You do NOT have to submit it

- Write your own function for computing conditional entropy in R. New function `entropy.cond(x,y)` will take two factors of the same length and will compute  $H(x|y)$ .

Example use: `entropy.cond(examples$SENSE, examples$A4)`

**You should understand and be able to explain and practically use**

- entropy
  - motivation
  - definition
  - main properties
  - calculation in R
- conditional entropy
  - definition and meaning
  - relation to mutual information
  - calculation in R
  - information gain – application in feature selection