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

- **Informal intro to Machine Learning**
- **Formal definition of Machine Learning**
    - Supervised Machine Learning
    - Features and target values
    - Prediction function
    - Loss function
    - Training and test data
    - Development cycle
- **Entropy**
- **Organizational notes**
    - Brief overview of the course
    - Credit and examination requirements
- **Summary of the lecture**

Informal explanation – motivation examples

Word-sense disambiguation (WSD)
Assign the correct sense of a word in a sentence.
Let's work with the word *line*:

- I've got Inspector Jackson on the **line** for you.
- Outside, a **line** of customers waited to get in.
- He quoted a few **lines** from Shakespeare.
- He didn't catch many fish, but it hardly mattered.
  With his **line** out, he sat for hours staring at the Atlantic.
- ...

Motivation example

Word-sense disambiguation

Assign the correct sense of a word in a sentence.

Let's work with the word *line* and its following senses:

- CORD
- DIVISION
- FORMATION
- PHONE
- PRODUCT
- TEXT

Motivation example — Word-sense disambiguation

?CORD **?DIVISION** **?FORMATION** **?PHONE** **?PRODUCT** **?TEXT**

- I've got Inspector Jackson on the **line** for you. `PHONE`
- Outside, a **line** of customers waited to get in. `FORMATION`
- He quoted a few **lines** from Shakespeare. `TEXT`
- He didn't catch many fish, but it hardly mattered.
  With his **line** out, he sat for hours staring at the Atlantic. `CORD`
- The company has just launched a new **line** of small,
  low-priced computers. `PRODUCT`
- Draw a **line** that passes through the points P and Q. `DIVISION`
- This has been a very popular new **line**. `PRODUCT? FORMATION?`

Motivation example

Word-sense disambiguation

- What knowledge do you use to assign the senses?
- What are the keys for the correct decision?

Motivation example

- We – human beings – do word sense disambiguation easily using the **context in the sentence** and having our **knowledge of the world**.
- We want computers to master it as well.

Let's prepare examples and guide computers to learn from them.

That is Machine Learning!

Formal definition of ML by Mitchell (1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine learning needs examples

Intuitively we need a large set of recognized **examples** to learn the essential knowledge necessary to recognize correct output values. Examples used for learning are called **training data**.

sentence	sense
I've got Inspector Jackson on the **line** for you.	PHONE
Outside, a **line** of customers waited to get in.	FORMATION
These companies rent private telephone **lines**.	PHONE
Please hold the **line**.	PHONE
He quoted a few **lines** from Shakespeare.	TEXT
He drew a **line** on the chart.	DIVISION
She hung the washing on the **line**.	CORD

What computers extract from examples

In the WSD task, both humans and computers need to know the **context of the target word** ("line") to recognize correct senses.

Humans use their reason, intuition, and their real world knowledge.

Computers need to extract a limited set of useful **context clues** that are then used for automatic decision about the correct sense.

- Formally, the context clues are called **attributes or features** and should be exactly and explicitly defined.
- Then each object (e.g. a sentence) is characterized by a list of features, which is called **feature vector**.

Computer makes feature vectors from examples.

Intuitive feature extraction – examples

To choose an effective set of features we always need our intuition. Only then all experiments with data can start.

A few example hints:

class	**a feature to recognize the class – will be useful?**
`CORD`	immediately preceding word
`FORMATION`	immediately following word
`PHONE`	can be often recognized by characteristic verbs

"Examples" in ML – two meanings

1) Real examples – Each real object that is already recognized or that we want to recognize is an example.

2) Data instances – In ML, each real example is represented as a data instance. In this sense

$$\textbf{example} = \textbf{feature vector} + \textbf{output value}$$

Data instances

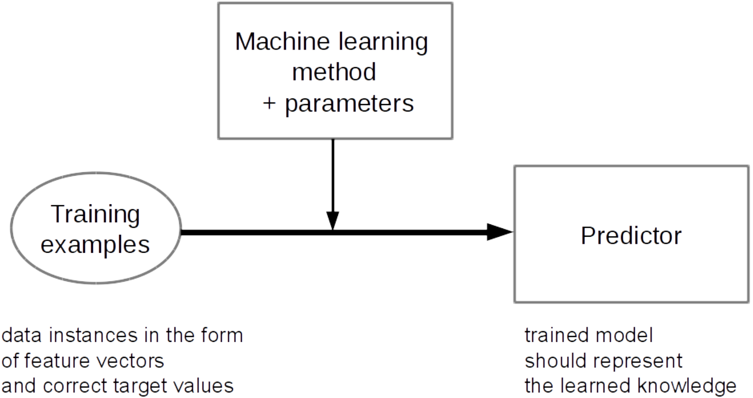
Sometimes we do not know the output value; in this case data instances are not different from feature vectors.

data instance = feature vector (+ output value, if it is known)

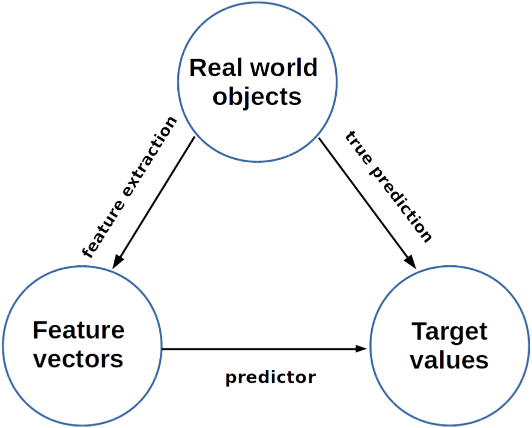
A data instance is either a feature vector or a complete example.

Supervised learning process

Supervised Machine Learning = computer learns "essential knowledge" extracted from a (large) set of examples



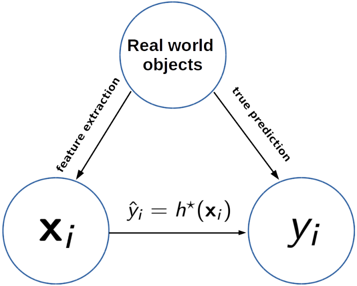
Machine learning as building a prediction function



- if target values are *continuous* numbers, we speak about **regression**
  = estimating or predicting a continuous response
- if target values are *discrete/categorical*, we speak about **classification**
  = identifying group membership

Prediction function and its relation to the data

Idealized model of supervised learning



- $\mathbf{x}_i$ are **feature vectors**, y_i are true **predictions**
- **prediction function** $\hat{f}^\star$ is the "best" of all possible hypotheses $\hat{f}$
- **learning process** is searching for $\hat{f}^\star$, which means to search the **hypothesis space** and minimize a predefined **loss function**
- ideally, the learning process results in $\hat{f}^\star$ so that predicted $\hat{y}_i = \hat{f}^\star(\mathbf{x}_i)$ is equal to the true target values y_i

Loss function

A loss function $\mathrm{L}(\hat{y}, y)$ measures the cost of predicting $\hat{y}$ when the true value is y. Commonly used loss functions are

- squared loss $\mathrm{L}(\hat{y}, y) = (\hat{y} - y)^2$
  for regression
- zero-one loss $\mathrm{L}(\hat{y}, y) = \mathrm{I}(\hat{y} \neq y)$
  for classifiation; *indicator variable* I is 1 if $\hat{y} \neq y$, 0 otherwise

The goal of learning can be stated as producing a model with the smallest possible loss; i.e., a model that minimizes the average $\mathrm{L}(\hat{y}, y)$ over all examples.

Important notes

- Loss function is sometimes also known as "cost function".
- In a broader sense, loss function means the value that summarizes the loss over a sample of examples, e.g. $\sum \mathrm{L}(\hat{y}, y)$ or $\mathsf{E}\,[\mathrm{L}(\hat{y}, y)]$.
- A more general term is "objective function", which is sometimes used for the function that should be optimized (minimized or maximized); yes, typically the objective function is in fact the loss function computed over a sample of development test examples.

Training data vs. test data

- **Training data** = a set of examples
  – used for **learning process**
- **Test data** = another set of examples
  – used for **evaluation** of a trained model
- **Important**: the split of all available examples into the training and the test portions should be **random**!

Supervised ML task and data instances

Supervised machine learning necessarily requires learning examples

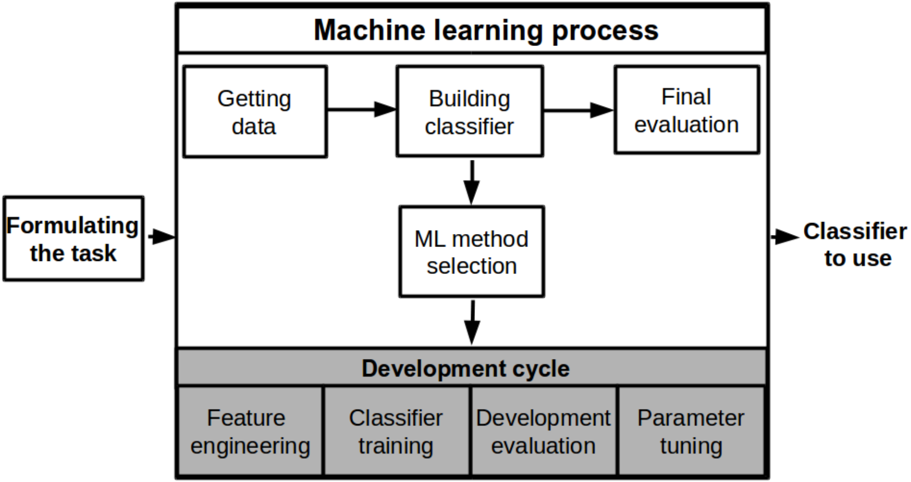
- **Features** are properties of examples that can be observed or measured
  – are numerical (discrete or continuous), or categorical (incl. binary)
- **Feature vector** is an ordered list of selected features
- **Data instance** = feature vector (+ target class, if it is known)
- **Training data** = a set of examples used for **learning process**
- **Test data** = another set of examples used for **evaluation**

Terminology – features and target values

- **How different people call values that describe objects**

	observed (known) object characteristics	values or categories to be predicted
computer scientists	**features**	**(target) value or class**
mathematicians (statisticians)	attributes or predictors	response (value) or output value

Machine learning process — development cycle



Terminological notes on building predictors

The purpose of the learning process is search for the best parameters of prediction function. – These parameters are the output of learning algorithms.

learning parameters (aka hyperparameters)	hypothesis parameters
= parameters of learning algorithm	= parameters of prediction function

- **Method** = approach/principle to learning. i.e. to building predictors
- **Model** = method + set of features + learning parameters
- **Predictor** = trained model, i.e. an output of the machine learning process, i.e. a particular method trained on a particular training data.
- **Prediction function** = predictor (used in mathematics). It's a function calculating a response value using "predictor variables".
- **Hypothesis** = prediction function – not necessarily the best one (used in theory of machine learning).

Practical procedures in the ML process

- **Formulating the task**
- **Getting data, examples**
- **Data preprocessing and feature extraction/selection**
- **Learning and evaluation**
- **Model assessment**

Formulating the task

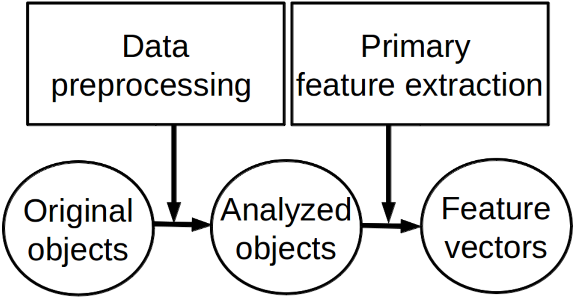
1. **Task description**
   WSD: Assign the correct sense to the target word "line"

2. **Object specification**
   WSD: Sentences containing the target word

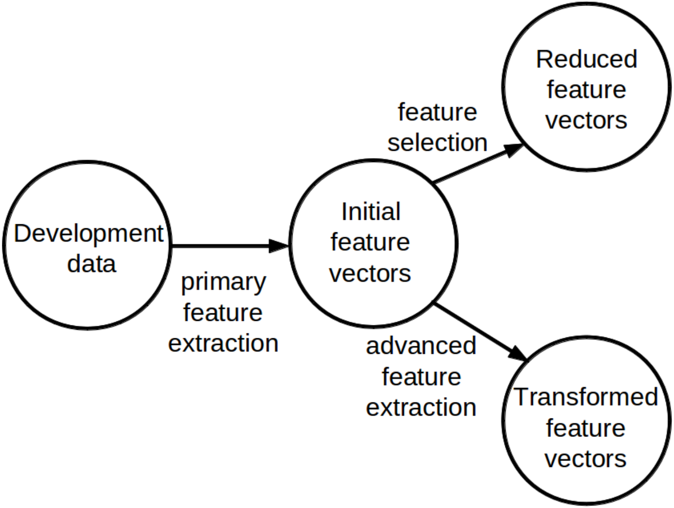
3. **Specification of desired output Y**
   WSD: $Y = \texttt{SENSE}$
   `SENSE = {CORD,DIVISION,FORMATION,PHONE,PRODUCT,TEXT}`

Data preprocessing and feature extraction

Step 1: Getting feature vectors



Feature extraction and feature selection



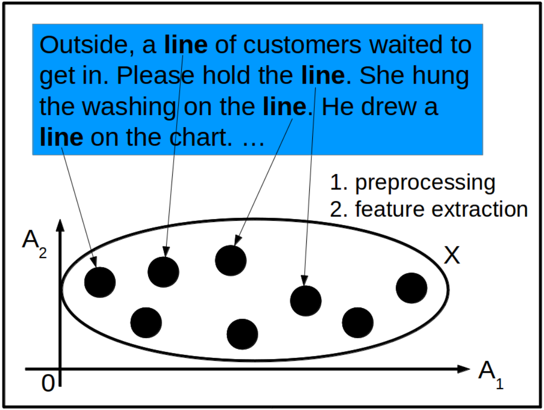
Getting data

Step 1: Getting feature vectors

- Features as variables A_1, ..., A_m
  - **numerical**
    – either discrete or continuous
  - **categorical**
    – any list of discrete values, non-numerical
  - **binary** (0/1, True/False, Yes/No)
    – can be viewed as a kind of categorical
- Feature values $x_1, ..., x_m$, $x_i \in A_i$
- Each object represented as feature vector $\mathbf{x} = \langle x_1, ..., x_m \rangle$
- Feature vectors are elements in an m-dimensional feature space
- Set of instances $X = \{\mathbf{x} : \mathbf{x} = \langle x_1, ..., x_m \rangle, x_i \in A_i\}$.

Getting data

Step 1: Getting feature vectors – Example



Example feature vectors – the WSD task

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
1	0	0	0	0	0	0	0	0	0	0	safety	special	install	inside	NN	IN	DT	lines	dobj
0	1	0	0	0	0	0	0	0	0	0	class	across	reach	.	NN	.	X	lines	prep_across
0	1	0	0	0	0	0	0	1	0	0	fine	the	walk	between	JJ	IN	JJ	line	dobj
0	1	0	0	0	0	0	0	1	0	0	fine	``	a	between	JJ	IN	VBG	line	dobj
0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
0	0	1	0	0	0	0	0	0	0	0	long	when	,	of	JJ	IN	NNS	lines	nsubj
0	0	1	0	0	0	0	0	0	0	0	long	in	patiently	to	JJ	TO	VB	lines	prep_in
0	0	1	0	0	0	0	0	0	0	0	long	the	but	delay	JJ	VBD	DT	lines	nsubj
0	0	0	0	1	0	0	0	0	0	0	car	the	X	affect	NN	VBN	IN	lines	nsubj
0	0	0	0	0	0	0	0	0	0	0	establish	of	marketing	such	VBN	JJ	IN	lines	prep_of
0	0	0	0	0	0	0	0	0	0	1	main	few	a	and	JJ	CC	RB	lines	prep_on
0	0	0	0	1	0	0	0	0	0	0	computer	new	the	to	NN	TO	VB	line	dobj

See the feature description `wsd.attributes.pdf` at
`https://ufal.mff.cuni.cz/course/npfl054/materials`

Getting data

Step 2: Assigning true predictions

- Take *a number* of original objects and assign true prediction to each of them, e.g. **do manual annotation**.
- Take these objects and their true prediction, do preprocessing and feature extraction. It results in **Gold Standard Data**

$$Data = \{\langle \mathbf{x}, y \rangle : \mathbf{x} \in X, y \in Y\}.$$

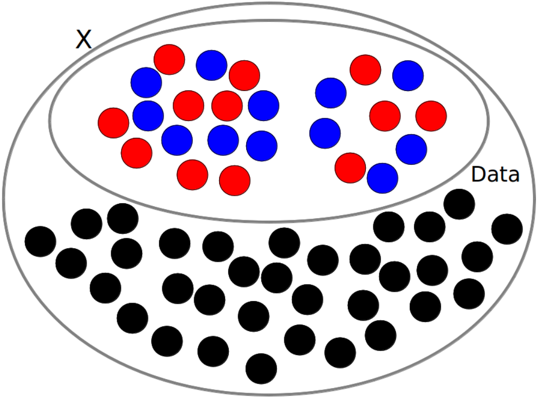
Getting data

Step 2: Assigning true prediction
Example: Y = SENSE = {CORD,DIVISION,FORMATION,PHONE,PRODUCT,TEXT}

SENSE	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
cord	1	0	0	0	0	0	0	0	0	0	0	safety	special	install	inside	NN	IN	DT	lines	dobj
division	0	1	0	0	0	0	0	0	0	0	0	class	across	reach	.	NN	.	X	lines	prep_across
division	0	1	0	0	0	0	0	0	1	0	0	fine	the	walk	between	JJ	IN	JJ	line	dobj
division	0	1	0	0	0	0	0	0	1	0	0	fine	``	a	between	JJ	IN	VBG	line	dobj
division	0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
division	0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
formation	0	0	1	0	0	0	0	0	0	0	0	long	when	,	of	JJ	IN	NNS	lines	nsubj
formation	0	0	1	0	0	0	0	0	0	0	0	long	in	patiently	to	JJ	TO	VB	lines	prep_in
formation	0	0	1	0	0	0	0	0	0	0	0	long	the	but	delay	JJ	VBD	DT	lines	nsubj
product	0	0	0	0	1	0	0	0	0	0	0	car	the	X	affect	NN	VBN	IN	lines	nsubj
product	0	0	0	0	0	0	0	0	0	0	0	establish	of	marketing	such	VBN	JJ	IN	lines	prep_of
product	0	0	0	0	0	0	0	0	0	0	1	main	few	a	and	JJ	CC	RB	lines	prep_on
product	0	0	0	0	1	0	0	0	0	0	0	computer	new	the	to	NN	TO	VB	line	dobj

Getting data

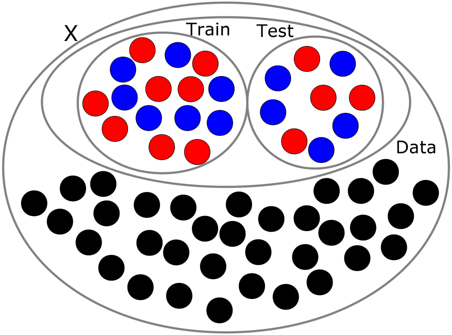
Step 2: Assigning true prediction
Example: $Y = \{red, blue\}$



Getting data

Step 3: Selecting training set *Train* and test set *Test*

- $Train \subseteq Data$, $Test \subseteq Data$
- $Train \cap Test = \emptyset$
- $Train \cup Test = Data$

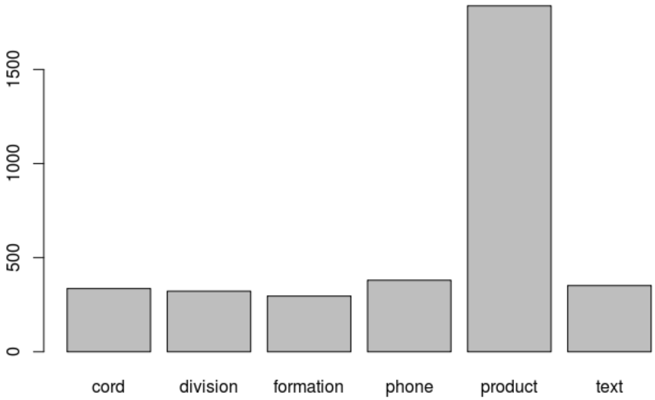


Entropy — uncertainty of random variables

- **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

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

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

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

This is what information theory calls entropy.

- Entropy of a random variable X is denoted by $\mathrm{H}(X)$
  – or, $\mathrm{H}(p_1, p_2, \ldots, p_n)$ 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

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

Continuity

$$\mathrm{H}(p, 1-p) \text{ is a continuous function}$$

Non negativity and maximality

$$0 \leq \mathrm{H}(p_1, p_2, \ldots, p_n) \leq \mathrm{H}(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$$

Symmetry

$$\mathrm{H}(p_1, p_2, \ldots, p_n) \text{ is a symmetric function of its arguments}$$

Recursivity

$$\mathrm{H}(p_1, p_2, p_3, \ldots, p_n) = \mathrm{H}(p_1 + p_2, p_3, \ldots, p_n) + (p_1 + p_2)\mathrm{H}(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2})$$

Entropy of SENSE

Entropy of `SENSE` **is** 2.107129 **bits.**

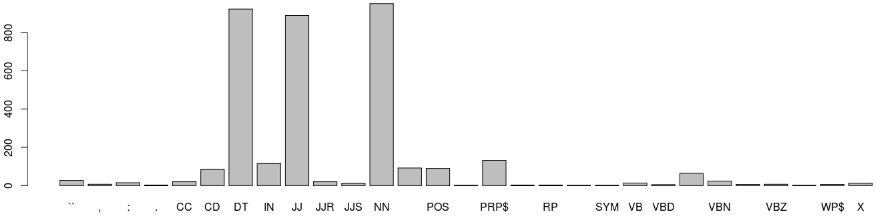
```
### probability distribution of SENSE
> p.sense <- table(examples$SENSE)/nrow(examples)
>
### entropy of SENSE
> H.sense <-  - sum( p.sense * log2(p.sense) )
> H.sense
[1] 2.107129
```

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
```

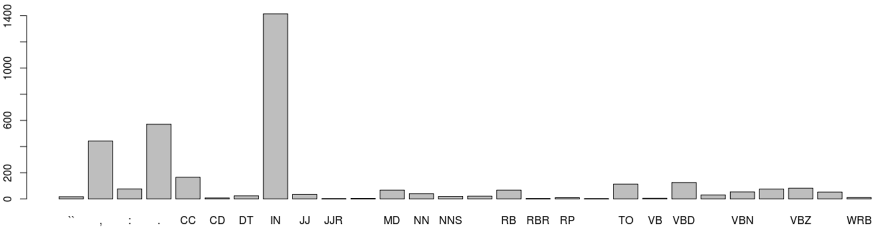
Distribution of feature values – A16

```
> 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)
>
```



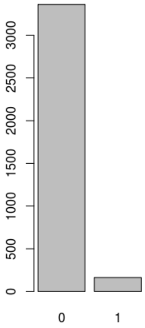
Distribution of feature values – A17

```
> 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
```

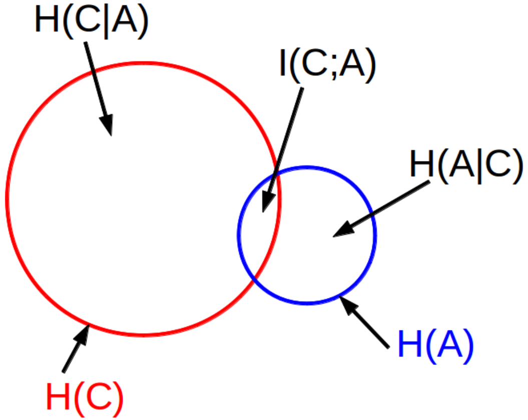
Conditional entropy H(C | A)

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

The answer is conditional entropy:

$$\boxed{\mathrm{H}(C \mid A) \quad = \quad - \sum_{y \in C, x \in A} \Pr(y, x) \cdot \log_2 \Pr(y \mid x)}$$

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

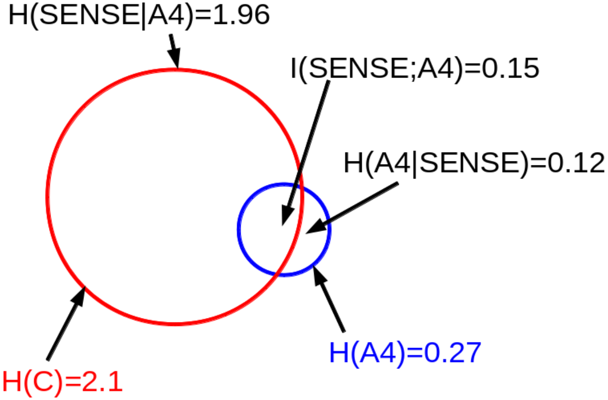
Mutual information measures the amount of information that can be obtained about one random variable by observing another.

Mutual information is a symmetrical quantity.

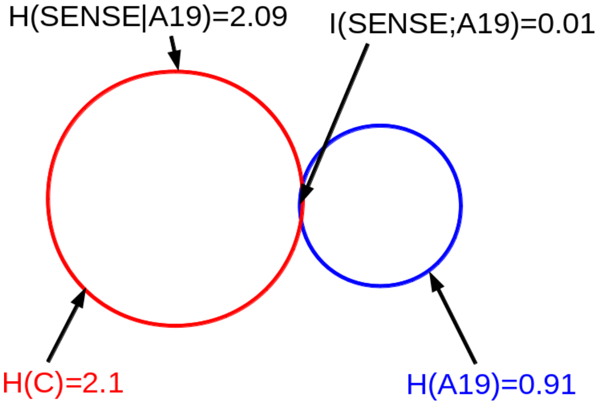
$$\boxed{\mathrm{H}(C) - \mathrm{H}(C \mid A) \quad = \quad \mathrm{I}(C; A) \quad = \quad \mathrm{H}(A) - \mathrm{H}(A \mid C)}$$

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

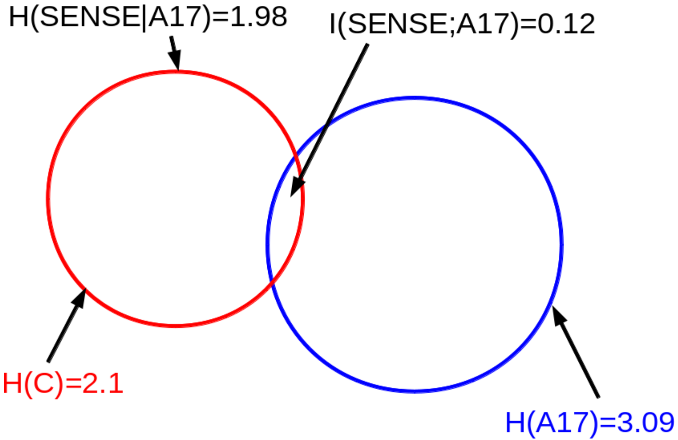
Conditional entropy – feature A4



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
```

Conditional entropy and feature selection

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 $\mathrm{H}(C \mid A)$, the better chance that A is a useful feature.
- However, since features typically interact, conditional entropy $\mathrm{H}(C \mid A)$ 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.

Organizational notes on the course

- **Two parallel classes – identical content**
- **Brief overview of the course**
  - This is an introductory course
  - We teach general foundations of ML
  - Main topics correspond to the exam requirements
- **Recommended literature**
  - ***An Introduction to Statistical Learning***
    by James, Witten, Hastie, and Tibshirani.
    Springer, New York, 2013. (available online)
  - ***Machine learning with R***
    by Brett Lantz.
    Packt Publishing Ltd. 2013. (available in the MFF library)

What you cannot learn in this course

- **no advanced methods**
  $\longrightarrow$ NPFL 097 Selected Problems in Machine Learning
- **no deep learning**
  $\longrightarrow$ NPFL 114 Deep Learning
- **no very details on Neural Networks**
  $\longrightarrow$ NAIL 002 Neural Networks
- **no special applications**
  $\longrightarrow$ e.g. NDBI 023 Data Mining
- **no advanced theoretical aspects of ML**
  $\longrightarrow$ NAIL 029 Machine Learning
- **no Weka, no Python libraries, etc.**
  - interested in Python?
    $\longrightarrow$ NPFL 104 Machine Learning Methods
    $\longrightarrow$ NPFL 129 Machine Learning for Greenhorns
    - a new course, very similar topics, exercises in Python

Machine Learning Lab Sessions

Goals of the lab sessions

- to learn how to practically analyse example data and ML tasks
- to learn how to practically implement some ML methods
- to solve a particular task
- practical experience with R system for statistical computing and graphics

  `http://www.r-project.org/`

Why statistics and probability theory?

Motivation

- In machine learning, models come from data and provide insights for understanding data (unsupervised classification) or making prediction (supervised learning).
- A good model is often a model which not only fits the data but gives good predictions, even if it is not interpretable.

Statistics

- is the science of the collection, organization, and interpretation of data
- uses the probability theory

Gentle introduction to R

What is R?

- a library of statistical tools
- an interactive environment for statistical analyses and graphics
- a programming language
- a public free software derived from the commercial system S

R is becoming more and more popular especially for its

- effective data handling and storage facility
- large, coherent, integrated collection of tools for data analysis
- well-developed, simple and effective programming language

Recommended reading

- *An Introduction to R*
  by W. N. Venables, D. M. Smith and the R core team
- also, an introduction available on the web:
  `http://cran.r-project.org/doc/manuals/R-intro.html`
- *R for Beginners* by Emmanuel Paradis

Supportive course NPFL 081
Practical Fundamentals of Probability and Statistics

- Intended and designed for students with weaker mathematical background
- We will go through **basics of probability theory and statistics**
- We will do **practical exercises using R system**
- Taught by Martin Holub and flexible for students' needs

Send a message to Holub@UFAL if you want to attend

Conditions for getting the credits

- **Obligatory Homeworks**
- **Written Tests**

  Scored Homeworks and written Tests are necessary conditions for attending the oral exam!

- **Oral examination requirements**

For more details see the course web page

Summary of Lecture #1
Examination Requirements

You should be familiar with key machine learning terms

- Machine learning process
- Development cycle
- Examples, feature vectors, data instances
- Gold standard data, training data, test data
- Manual annotation (true predictions)
- Model, predictor, hypothesis optimization
- Supervised learning
- Classification, regression
- Entropy, its meaning and basic definition – more details including conditional entropy will be discussed at the next lab session

Summary of Lecture #1
Homework

- Install R on your own computer and get familiar with its basic functions

What you will learn at the following Lab session #1

- **Annotation experiment**
  - Practical experience with manual annotation
- **Startup with R**
  - Elementary data processing and computation in R
  - Annotation data analysis