Summary - Anno4ML

Leon F.A. Wetzel
l.f.a.wetzel@student.rug.nl

Version 1.1 - 16-06-2020

---

1. Introduction to annotating data

Why is annotation important?

- Machine learning algorithms work better with good, representative data.
- Data needs to be prepared, so that machine learning algorithms can learn easily.
    - Metadata can be a part of this.

Layers of linguistic descriptions

Syntax	How words and phrases are combined into phrases and sentences.
Semantics	Meaning of phrases/sentences and relations over these meanings.
Morphology	Smallest units of language that have meaning or function (also known as **morphemes**)
Phonology	Sound patterns of a particular language, also known as phonemes.
Phonetics	Sounds of human speech, and how they are made and perceived.
Lexicon	Words and phrases used in a language.
Discourse analysis	Exchanges of information and the flow of information across sentence boundaries.
Pragmatics	How the context of text affects the meaning of an expression and how to recover a hidden/presupposed meaning.
Text structure analysis	How narratives and other textual styles are constructed to make larger textual compositions.

What can you do with Natural Language Processing?

Machine translation	Automatically translating from one language to another.
Speech recognition	Convert speech to text.
Question answering	Give answers based on a knowledge database.
Summarization	Produce a coherent summary from collections.
Document classification	Identify an application-dependent category of a document.
Fact checking	Check the correctness of a given statement.
Segmentation	Detect token or sentence boundaries.
POS-tagging	Guess a grammatical category of a token.
Syntactic parsing	Recognize a sentence and assign an underlying syntactic structure to it.
Word sense disambiguation	Detect a sense of a word.
Semantic parsing	Assign a structure to a sentence that reflects its (approximate) meaning.
Natural language inference	Identify semantic relation between meanings of natural language sentences.

What are types of annotation?

Part of speech
Token-based tagging, serving as an initial word sense disambiguation.

- Can be used in conjunction with **syntactic bracketing**.
- Example: John (NOUN) eats (VERB) soup (NOUN)

Semantic typing
Token/phrase-based tagging, denoting a type from a reserved vocabulary or ontology.

- Examples include:
    - MUC Message Understanding Conference
    - ACE Automatic Content Extraction

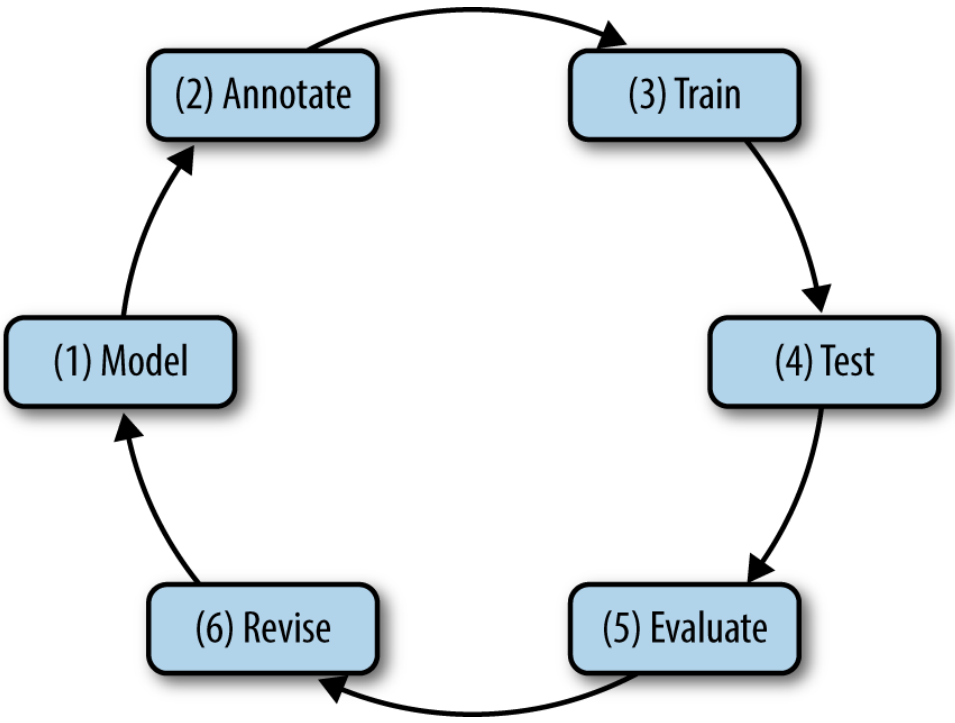
Semantic role labeling
Token/phrase-based tagging. A token/phrase (in a sentence) gets a specific semantic role (from a fixed list).

- Examples include:
  - [The man]$_{agent}$ painted [the wall]$_{patient}$ with [a paint brush]$_{instrument}$.
  - [My brother]$_{theme}$ lives in [Milwaukee]$_{location}$.
- Examples of semantic role lists:
  - VerbNet;
  - FarmeNet;
  - PropBank.

Learning from data

Supervised learning	• Learn a hypothesis that maps inputs to a predefined structure or a sequence/set of labels. • Needs annotated data for training.
Unsupervised learning	• Learn structure from unlabeled data. • Does not need annotated data to train.
Semi-supervised learning	• Combines aspects from supervised and unsupervised learning. • Also known as hybrid learning.

The MATTER cycle



Model
Design the phenomenon. A model consists of:

- **Terms**: annotation types, the actual labels that could be assigned.
- **Relations**: how the labels relate to each other.
- **Interpretation**: what do the labels mean, what is their meaning?

Annotate
Annotate the data with the specification. This part takes into account:

1. Design specification
    a. Consuming tags
    b. Non-consuming tags
2. Write guidelines
    a. Span of the tag
    b. Degree of interpretation
3. Annotate!
4. Evaluate and revise (if needed)
    a. Understanding or just agreement
5. Adjudication and gold standard

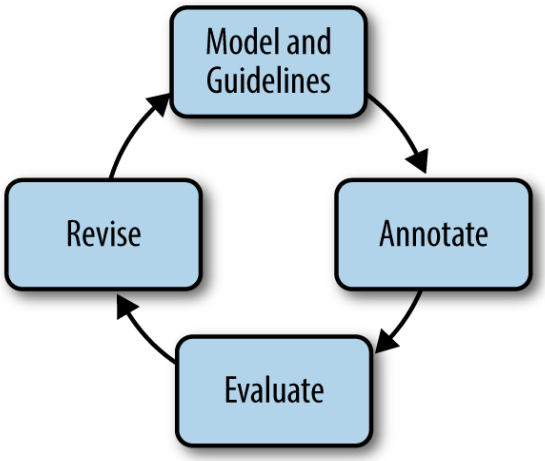
Train, Test and Evaluate
Apply the model on real data and look at the results. Division of the dataset usage is roughly:

- 60% = Training
- 25% = Testing
- 15% = Dev-testing

Revise
Apply changes to the annotation procedures, for example:

- Introduce a new tag or type
    - Example: add an Event type
- Split an existing tag or type
    - Example: Distinguish geopolitical from non geopolitical
- Collect more data

After **Revise**, the MATTER cycle restarts!



The MAMA cycle

Slight alteration to the MATTER cycle. MAMA stands for **Model-Annotate-Model-Annotate**. Used for the development of the annotation model and guidelines.
A part of the corpus is annotated according to these guidelines.

Annotation goals

Why do you want to annotate data?
A statement of purpose includes questions like…

- What will the annotation be used for?
    - Example: databases
- What will the overall outcome of the annotation be?
    - Example: genre classification
- Where will the corpus come from?
    - Example: news articles
- How will the outcome be achieved?
    - Example: usage of keywords

Example statement of purpose
I want to use keywords to detect the genre of a newspaper article in order to create databases of categorized texts.

Refining the goal
There is always room for improvement. Make annotation easy for annotators: if the task gets easier, less mistakes tend to arise. However, easy tasks could limit the actual value of annotations in late stages of processing.

Informativity versus correctness

- Informativity: annotation that is most useful for your task.
- Correctness: annotation that is not too difficult for annotators to complete accurately.

Scope of annotation
How far-reaching is the annotation goal?

- What relations?
- Only main verb per sentence?
- Intra/inter-sentence relations?
- Only events with clear temporal anchors?
- **How specific are the categories?**

Source of the corpus
Where will the corpus come from?

- Include or exclude similar sources?
    - For example: RTV Noord versus Dagblad van het Noorden
- Are blogs news sources?
- Exclusively online newspapers?
- Only written articles or also transcripts of broadcasts?
- **Which publication styles?**

Data collection

How restrictive are the licences that the sources have?

- Can you use the data freely (for your specific goals)?
- Do you need to pay (yearly, monthly) for using the data?

How representative are your sources?

- Can you use multilingual corpora to train your system?
- Can a local corpus be used for national or international purposes?

How balanced is your data?

- Are the labels distributed equally?
- Are there labels that are underrepresented or overrepresented?

How do you collect data?

- Internet
    - Open data
    - Research data banks
    - API's
- Elicit from people: **experts** versus **crowd**

What is the ideal size of a corpus?

Really depends on several factors…

- Complexity of the annotation task
- Time
- Money
- Limited resources
- Type of NLP task

2. Standards and specifications

What are specifications?

Imagine that we have a model M, which has the form…
Model = <Terms, Relations, Interpretation>

Model

All tag and attribute information in XML.

Specification
A concrete representation of the model.

- DTD: a concrete information structure of XML files.

Generality versus specificity

How detailed should the labels be?

Too general?

- Annotators can get confused.
- Information can get lossed.

Too detailed?

- Annotators cannot annotate properly or accurately.
- Annotated data can get unreliable quickly.

It is important to find the right modus!

Using existing models and specifications
A lot of tested assets and techniques are available for usage!

- Annotation software (brat)
- Data (annotated corpora)
- Annotation guidelines (NLTK standard tags)

How?

- Make them more specific to your context.
- Borrow ideas from others.

Standards

Don't reinvent the wheel, use what is available!

How do you apply and adopt annotation standards?

- What do you want your annotators to do?
- How do you want them to do it?
- How easily accessible is your data?

Document-level annotation

Annotating whole documents, instead of its individual contents separately.

- Labels are added to filenames

Text extent annotation

Sections of the text are given distinct labels (can span over words). Think of...

- POS-tagging
- Word sense disambiguation
- Named entity

How?

- Inline annotations
- Stand-off: location in a sentence
- Stand-off: character location

Inline annotation

- Widely used
    - Mainly due to its simplicity
- Annotations are not always easy to read.
- Changes original text.
- Impractical for combining more than one annotations.

Stand-off annotation with tokens

1. Tokenize the text.
2. Every token has a coordinate and location in the text.

- Token spanning tags are allowed.
- It is not easy to recover the original data.
- It is tough to pair annotation data and the original data.
- Not suitable for morpheme-level annotation.

Stand-off annotation with characters

- Read text character by character.
- Character positions are coordinates for location in text.
    - Depends on character encoding.
- Token spanning tags are allowed.
- Original data is intact.
    - A document can have multiple annotation layers.

Linked extent annotation

- Defined over IDs, instead of extents in the text.

Token-based annotation layers

Segmentation
Split texts into sentences, and split sentences into tokens (meaningful atoms/words).

Symbolization
Map tokens to non-logical symbols

- Lemmatization: ideal for morphological analysis, reducing types
  - *Is* becomes *be*
  - *Doors* become *door*
- Normalization: mapping to a canonical form
  - *Third* becomes *3*
  - *Played* becomes *play*
  - *Km* becomes *kilometer*

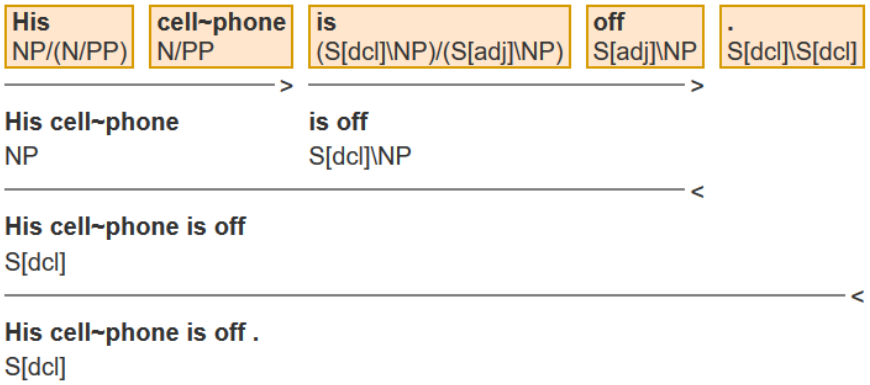
Word sense disambiguation
Assigning sense numbers to non-logical symbols.
Not all symbols get a sense number.

- Noun concepts
  - Named entities
  - Pronouns (gender)
- Verb concepts
- Adjective concepts
- Adverb concepts

token	symbol	sense
third	3	O
men	man	man.n.02
played	play	play.v.02
2:30 pm	14:30	O
Kraft	kraft	company.n.01

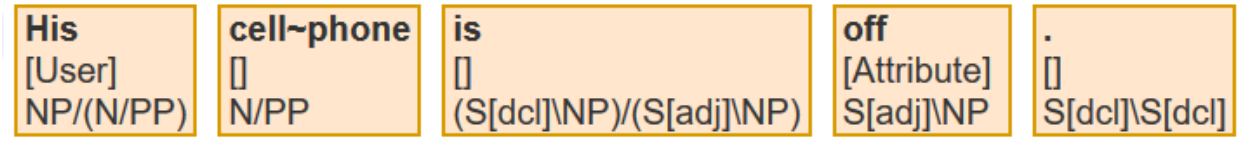
Syntactic parsing
Analyzing a string of symbols conforming to the rules of a formal grammar.

- Goes well with compositional semantics
- Lexicalized grammar
- Efficient and wide-coverage parsers are available.



Semantic roles
How do tokens relate to each other?

- The roles are mainly borrowed from VerbNet.
- Only tokens with function categories can have roles.



3. Guidelines for annotation

Preparing the annotation process

Guidelines versus specifications

- Specification: a concrete representation of an annotation goal.
- Guidelines: how to apply an annotation specification to the data.

Annotation schema
How to format the annotations.

1. Convert the data to a friendly format that is compatible with the annotation tool of choice.
2. Decide which information you would like to present to your annotators.
    - Metadata
        i. Document sources
    - Pre-marked up
        i. Necessary for annotators?
        ii. Is the info correct?
            1. Too much information can lower the annotators's accuracy.
3. Apply patches (if needed)
    - Split the files
    - Revise the guidelines

How do you write annotation guidelines?
The guidelines are the instructions for annotators how to apply annotation specifications (the schema) to the data.

What do you include in the guidelines?

- What is the goal of the project?
- What is each tag called and how is it used?
- What parts of the text do you want annotated, and what should be left alone?
- How will the annotation be created?

Using the guidelines

Single label usage

- What is the goal of the project?
    - *Label movie reviews as being positive or negative.*
- What is each tag called and how is it used?
    - *Positive OR Negative*
    - *Positive is assigned to positive reviews.*
    - *Negative is assigned to negative reviews and neutral ones.*
- What parts need to be annotated and what should be left alone?
    - *Label the entire document with a single label.*
- How will the annotation be created?
    - *Type the label next to the file name of the review in the spreadsheet.*

Multiple label usage

- What is the goal of the project?
    - *Label movie summaries with genre notations.*
- What is each tag called and how is it used?
    - *26 genre tags can be applied to each summary as needed.*
    - *Action, Adventure, News, History…*
- What parts need to be annotated and what should be left alone?
    - *Each label applies to the entire document.*
- How will the annotation be created?
    - *Annotation software X to apply multiple labels to a document.*

Given the guidelines for multiple label usage…

- What is the maximum number of labels per document?
    - *1? 3? Unlimited?*
- When do you apply which tag?
    - *What is the difference between Adventure movies and Action movies?*
- Not all genres describe the same aspect.

Extent annotations

What is the scope of each tag?

- The Netherlands, The Hague
- Neymar Jr., John F. Kennedy, Dr. Watson
- Frank and Ronald de Boer
- Elon Musk, the CEO of SpaceX
- Café The Crown

- Guidelines with two parts
    - Definition of tags

    - Tricky cases
- Possible several tags per named entity
    - Writer and director
    - Professor and researcher and rector magnificus
- Defining scope of tags has a great impact on annotator agreement

Link tags

- What are the links connecting?
    - Semantic roles?
    - Relationships?
- When should a link be created?

Annotators and the annotation environment

Which languages do annotators need to know?

- Close reading of a text by a native speaker
- Second-language learners

Does the annotation task require specialized knowledge?

- POS-tagging
    - A course on syntax could help!
- Tagging domain-specific concepts
    - Genes in biomedical papers
    - Juridical terms and concepts

What practical matters need to be taken into account?

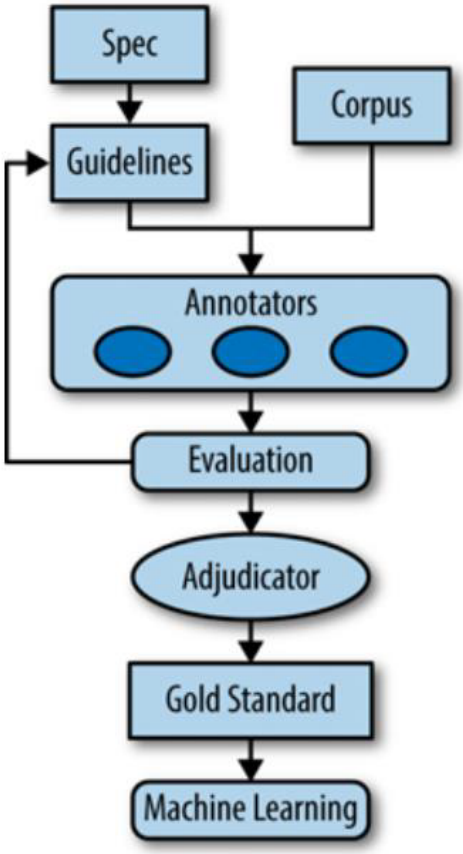
- Annotation time
    - Token-based versus document-level
- Data size and the amount of annotators
- Finances
    - Crowdsourcing
    - Trained annotators

Annotation environment

Facilitates the annotation procedures.

- From simple annotation tools to workbenches
    - Think of brat
- Annotation environments are not one-tool-fits-all
- Key factors in choosing the right environment:
    - How well are they supported
    - Availability for common OSes
    - Open source (with documentation)
    - Support for specific features

- Link tags
- Units of annotation
- Annotation layers
- Task management
- Adjudication



4. Inter-Annotator Agreement

Introduction

What is the Inter-Annotator Agreement? It shows…

- How clear the guidelines are
- How uniformly the annotators understood the task
- How reproducible the annotation task is

High IAA does NOT automatically lead to higher performing machine learning algorithms!
But large amounts of (high-quality) data could increase the performance of machine learning models.

Using accuracy for measuring IAA

We use the table below to calculate the IAA, using accuracy. In total, **24** annotations (N) have been made. In **10** cases, there was agreement between the two annotators and in 14 cases there was disagreement. Note that the ratios are rounded where applicable. 0,4167 is the ratio Agreement / N.

10		**Annotator B**		
	24	**Positive**	**Negative**	
Annotator A	**Positive**	3	9	12
	Negative	5	7	12
		8	16	0,4167

Now we calculate the distributions...

$A_{positive}$ / N	$A_{negative}$ / N	$B_{positive}$ / N	$B_{negative}$ / N
0,5	0,5	0,3	0,6667

Now we can calculate…

$$IAA = A_{negative} \cdot B_{negative} + A_{positive} \cdot B_{positive} = 0.5 \cdot 0.667 + 0.5 \cdot 0.3 = 0.4833$$

Using the F1 score for measuring IAA

Calculating the F1-score, precision and recall

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \qquad Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

10		Annotator B		
	24	**Positive**	**Negative**	**Precision**
Annotator A	**Positive**	3	9	0,25
	Negative	5	7	
	Recall	0,375		0,4167

$$Precision = \frac{TP}{TP + FP} = \frac{3}{3 + 9} = 0,25 \qquad Recall = \frac{TP}{TP + FN} = \frac{3}{3 + 5} = 0,375$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} = \frac{2 \cdot 0,25 \cdot 0,375}{0,25 + 0,375} = \frac{0,1875}{0,625} = 0,3$$

Introducing kappa

Why use kappa instead of F1 and accuracy?
F1 and accuracy don't take into account expected chance agreements that are likely to occur when people annotate texts!

Cohen's κ	Two annotators annotating each subject with a category.
Fleiss' κ	Each subject was annotated N times with a category. ● Is a generalized version of Scott's π ● Can work for three annotators and more ● N: does not mean by the same annotators!

How do you calculate kappa? $$\kappa = \frac{A_0 - A_C}{1 - A_C}$$

Where:

- A_0 = Observed agreement
- A_C = Chance agreement

Cohen's κ

$$\kappa = \frac{P_0 - P_C}{1 - P_C}$$

There are 85 annotations (N), of which 48 are agreements (56,47%)..

48		Annotator B				
	85	Positive	Neutral	Negative		
Annotator A	Positive	23	6	8	37	**0,4352**
	Neutral	4	18	5	27	**0,3176**
	Negative	2	12	7	21	**0,247**
		29	36	20	0,5647	
		0,3411	**0,4235**	**0,2352**		

The observed agreement (P_O) can be calculated by dividing the amount of agreements by the total amount of annotations. This value is also our accuracy.

$$P_O = \frac{48}{85} = 0,5647$$

The distribution of categories per annotations can be calculated by dividing the sum of a row or column by the total amount of annotations. The chance agreement (P_C) can be calculated by multiplying the sum of (A_{pos} * B_{pos}), (A_{neu} * B_{neu}) and (A_{neg} * B_{neg}).

$\sum A_{pos} \cdot \sum B_{pos}$	$\sum A_{neu} \cdot \sum B_{neu}$	$\sum A_{neg} \cdot \sum B_{neg}$	P_C
0,1484	0,1345	0,0581	0,341

We can now calculate Cohen's κ!

$$\kappa = \frac{0,5647 - 0,341}{1 - 0,341} = \frac{0,2237}{0,659} = 0,3394$$

To summarise:

S		**B**					
	N	c_1	...	c_k			
A	c_1	n_{11}	...	n_{1k}	A_1	p_1^A	$p_1^A \times p_1^B$
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	c_k	n_{k1}	...	n_{kk}	A_k	p_k^B	$p_k^A \times p_k^B$
		B_1	...	B_k	$\boldsymbol{p_o = S/N}$	$\boldsymbol{1 - p_c}$	$\boldsymbol{p_c}$
		p_1^B	...	p_1^B	$\boldsymbol{p_o - p_c}$	$\boldsymbol{\kappa}$	

Fleiss' κ

$$\kappa = \frac{\bar{P}_O - \bar{P}_C}{1 - \bar{P}_C}$$

90 annotators annotate 4 movie reviews with three categories (positive, neutral and negative). A total of 360 annotations have been made.

360	**Positive**	**Neutral**	**Negative**		
Review 1	20	32	38	90	0,3468
Review 2	35	43	22	90	0,4329
Review 3	57	34	9	90	0,5488
Review 4	13	24	63	90	0,5772
	125	133	132		
Distribution of categories	0,3472	0,3694	0,3666	0,3914	

$$\frac{20^2 + 32^2 + 38^2 - 90}{90^2 - 90} = 0,3468 \qquad \frac{35^2 + 43^2 + 22^2 - 90}{90^2 - 90} = 0,4329$$

$$\frac{57^2 + 34^2 + 9^2 - 90}{90^2 - 90} = \frac{4396}{8010} = 0,5488 \qquad \frac{13^2 + 24^2 + 63^2 - 90}{90^2 - 90} = \frac{4624}{8010} = 0,5772$$

After we have calculated the distribution ratio per review, we can sum these values and divide them by the amount of reviews that we have…

$$\frac{0,3468 + 0,4329 + 0,5488 + 0,5772}{4} = 0,4764$$

$$\kappa = \frac{\overline{P}_O - \overline{P}_C}{1 - \overline{P}_C}$$

We now have our $\overline{P}_O$ value! We can now finally calculate Fleiss' kappa.

$$\kappa = \frac{0,4764 - 0,3914}{1 - 0,3914} = 0,1396$$

Scott's π

Is basically Fleiss' kappa for two annotators. Let's calculate this value. We have 10 reviews, of which in 7 cases there is agreement.

7		**Annotator B**	
	10	**Positive**	**Negative**
Annotator A	**Positive**	3	2
	Negative	1	4

	Positive	**Negative**		
Review 1	2	0	2	1
Review 2	2	0	2	1
Review 3	2	0	2	1
Review 4	1	1	2	0
Review 5	1	1	2	0
Review 6	0	2	2	1
Review 7	0	2	2	1
Review 8	0	2	2	1
Review 9	0	2	2	1
Review 10	1	1	2	0
	9	11		0,7
	0,45	0,55	0,505	0,3939

$$\frac{2^2 + 0^2 - 2}{2^2 - 2} = \frac{2}{2} = 1 \qquad \frac{1^2 + 1^2 - 2}{2^2 - 2} = \frac{0}{2} = 0 \qquad \overline{P}_O = 0,45^2 + 0,55^2 = 0,505$$

$$\overline{P}_C = \frac{1 + 1 + 1 + 0 + 0 + 1 + 1 + 1 + 1 + 0}{10} = 0,7$$

$$\kappa = \frac{0,7 - 0,505}{1 - 0,505} = \frac{0,195}{0,495} = 0,3939$$

Comparing Cohen's κ and Fleiss' κ

Cohen's κ	**Fleiss' κ**
2 annotators	X annotators
Each annotator annotates every item	Every item is not necessarily annotated by each (the same) annotator
Table: **confusion matrix** ● Annotation counts of Annotator 1 ● Annotation counts of Annotator 2	Table: **annotation counts per item** ● Items ● Annotation counts per item

Scott's π

Fleiss' κ is an extension of Scott's π for two annotators.

- Not for Cohen's κ!

Cohen's κ is more informative than Scott's π due to the wat chance agreement is calculated.

- Use Cohen's κ over Scott's π when you have two annotators.

Interpreting kappa coefficients

Values	**Agreement level**
$K \leq 0$	Poor
$0 < K \leq 0,2$	Slight
$0,2 < K \leq 0,4$	Fair
$0,4 < K \leq 0,6$	Moderate
$0,6 < K \leq 0,8$	Substantial
$0,8 < K \leq 1$	Almost perfect

$K = 1$	Perfect

5. Machine learning

What is learning?

Herbert Simon
"Learning is any process by which a system improves its performance from experience."

Tom Mitchell
"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."

What can you learn about language?

- POS-tagging
- Topic modelling
- Handwriting recognition
- Sentence parsing
- Coreference resolution
- Named entity identification

Target function
A function that maps input data to the desired output.

- The hypothesis (function) attempts to approximate the target function.
- **Hypothesis space**
  - A collection of all possible hypothesis functions
- **Learning from experience**
  - Learning from training examples
- Hypothesis overfits the training examples if some other hypothesis that fits the training examples less well actually performs better over the entire distribution of instances.

Learning task
Learning involves improving on task T with respect to a performance metric P, based on experience E.

1. Choose the training experience
2. Identify the target function
3. Decide how to represent the target function
4. Choose a learning algorithm
5. Evaluate the results with the performance metric

Feature selection

- N-grams
- Structure-dependent
    - Length
    - Nth element
    - Amount of vowels
- Annotation-dependent
    - New, explicitly added information that can help in classification or discrimination
        - Person
        - Organization
        - Place

Target functions

- Classification
    - Binary
        - Logistic regression
        - Bernoulli naive bayes
        - *Spam versus not-spam*
        - *Sentiment analysis*
    - Multi-class
        - Multinomial logistic regression
        - Multinomial naive bayes
        - *Natural language inference*
        - *Genre detection*
- Structure prediction
    - Sequence labeling
        - POS tagging
        - Segmentation
    - Parsing
        - Semantic parsing
        - Syntactic parsing
- Regression analysis
    - Scalar value
    - Linear regression
        - Property prices
        - Degree of similarity

Introducing entropy

What is entropy?

A measure of uncertainty, chaos, mess and diversity.

How do you calculate entropy?
For calculating the entropy of a discrete random variable X…

$$H(X) = -\sum_{x \in V(X)} p(x) \log p(x) = \sum_{x \in V(X)} p(x) \log \frac{1}{p(x)}$$

Where:

- V(X): values of the random variable
- p(x): probability mass function
- log: has base = 2, serves a a scale

<u>Entropy does not depend on the values of the random variable!</u>

Information gain

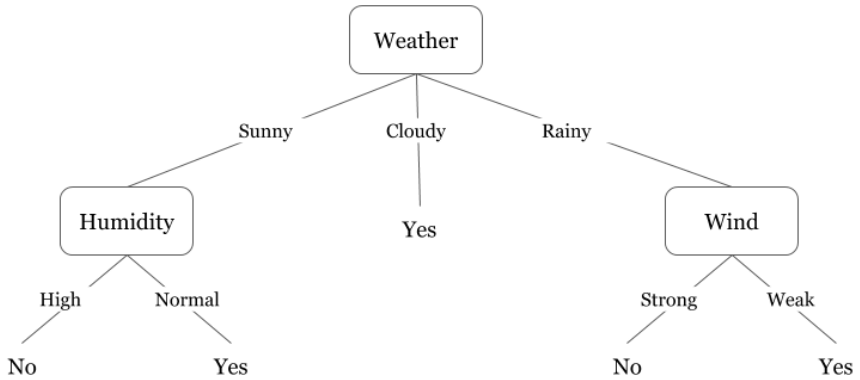
What is information gain?
The amount of information gained about a random variable from observing another variable.

$$Information\ gained = \overline{chaos\ before} - \overline{chaos\ now}$$

$$Gain(S,A) = H(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} H(S_v)$$

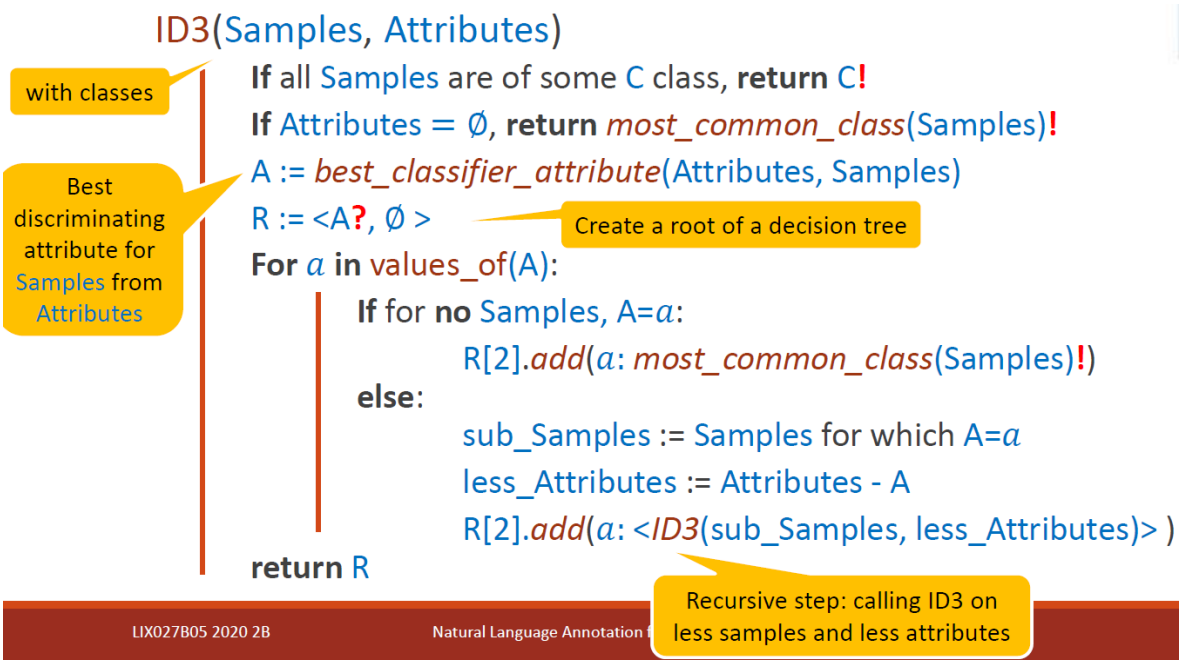
Where:

- Gain: information gain (with ID3, see next section!)
- H(S): entropy with respect to the target class
- $\frac{|S_v|}{|S|}$: weight



ID3 algorithm

Iterative Dichotomiser 3 algorithm
Used for generating a decision tree based on a given dataset.



The function **best_classifier_attribute** is synonymous with the information gain formula shown on the previous page. Let's execute this function with the data from the slides of week 4. We start off with calculating the information gain for the Outlook variable, in conjunction with the PlayGolf variable (as target variable!).

		Play Golf		
		Yes	**No**	
Outlook	**Sunny**	3	2	**5**
	Overcast	4	0	**4**
	Rainy	2	3	**5**
				14

Based off the table shown here, our information gain formula looks like…

$$Gain(S,A) = H(\{5,9\}) - -\frac{5}{14}H(\{3,2\}) - \frac{4}{14}H(\{4,0\}) - \frac{5}{14}H(\{2,3\})$$

		Play Golf		
		Yes	**No**	
Outlook	**Sunny**	3	2	**5**
	Overcast	4	0	**4**
	Rainy	2	3	**5**
				14

Our function H(S)...

$$H(X) = -\sum_{x \in V(X)} p(x) \log p(x) = \sum_{x \in V(X)} p(x) \log \frac{1}{p(x)}$$

Calculate the probability per row.

$$H(\{3,2\}) = \frac{3}{5} \cdot \log_2\left(\left(\frac{3}{5}\right)^{-1}\right) + \frac{2}{5} \cdot \log_2\left(\left(\frac{2}{5}\right)^{-1}\right) = 0,971$$

$$H(\{4,0\}) = \frac{4}{4} \cdot \log_2\left(\left(\frac{4}{4}\right)^{-1}\right) + \frac{0}{0} \cdot \log_2\left(\left(\frac{0}{0}\right)^{-1}\right) = 0$$

$$H(\{2,3\}) = \frac{2}{5} \cdot \log_2\left(\left(\frac{2}{5}\right)^{-1}\right) + \frac{3}{5} \cdot \log_2\left(\left(\frac{3}{5}\right)^{-1}\right) = 0.971$$

Calculate the entropy of PlayGolf and Outlook.

$$E(PlayGolf, Outlook) = \frac{5}{14} \cdot H(\{3,2\}) + \frac{4}{14} \cdot H(\{4,0\}) + \frac{5}{14} \cdot H(\{2,3\}) = 0,6936$$

Calculate the overall entropy of PlayGolf and Outlook

$$H(\{5,9\}) = \frac{5}{14} \cdot \log_2\left(\left(\frac{5}{14}\right)^{-1}\right) + \frac{9}{14} \cdot \log_2\left(\left(\frac{9}{14}\right)^{-1}\right) = 0,940$$

Calculate the information gain.

$$Gain(S,A) = H(\{5,9\}) - -\frac{5}{14} \cdot H(\{3,2\}) - \frac{4}{14} H(\{4,0\}) - \frac{5}{14} H(\{2,3\}) = 0,247$$

You can calculate the information gain in the same way for the other features. A decision tree can be constructed by sorting the features by their information gain, from highest to lowest.

6. Decision trees

Using ID3

ID3 can be used to construct decision trees

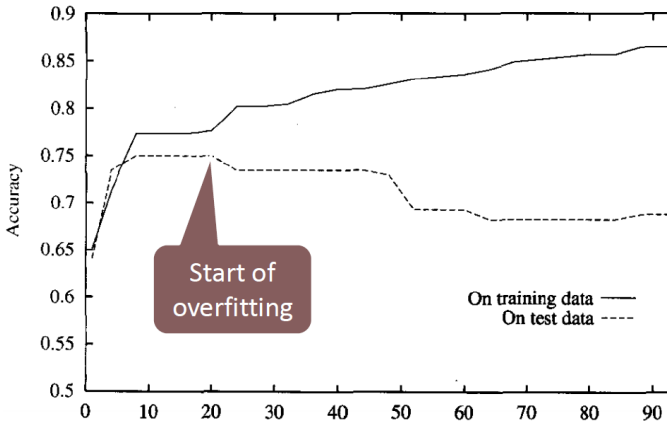
- ID3's hypothesis space is complete
  - Decision trees are expressive
- ID3 maintains only a single current hypothesis as it searches
  - No information about other decision trees that fit the training data
  - Unsupervised learning!
- No backtracking!
  - Does not return to a higher level or node
  - Search is incomplete, unless at final node
  - Prefers locally optimal solutions
  - Local optimal is not necessarily the global optimal
  - Greedy algorithm
- Inductive bias
  - Shorter trees are preferred over larger trees
- Uses all training examples at each step of building
  - More immune to noisy examples

Issues with trees

How deep should trees grow?

Trees with homogenous leaves could lead to overfitting data.

- Noisy examples can significantly affect the final tree.
- Few samples at the leaf nodes?
  - Coincidental regularities?
- How do you combat overfitting?
  - Stop the growth of a tree before the perfect fit.
  - Fit the data and post-prune the tree.



- Keep the tree preferably small!

Post-pruning

How do you do it?

- Split the training data
  - Training set (⅔)
  - Validation set (⅓)
- Fit the training set with a decision tree
- Prune the tree guided by the performance on the validation set
  - For each node N: prune N if it leads to improvements on the validation set
- **Drawback:** requires sufficient data for the split

Decision tree rules and post-pruning

- Fit the training data with a tree
- Convert the tree into a set of rules
- Prune (generalize) rules
  - Remove any preconditions that improve the performance
- For classification: sort the rules by their estimated accuracy
  - Either on training or development set
  - Sorting on accuracy leads to improved performance in the end

Why should you prune rules, instead of trees?

- Different contexts of the nodes can be distinguished
  - After splitting nodes, they can be differently treated
- Discard the attribute hierarchy
  - Pruning top attributes is easy
- Rules are more readable!

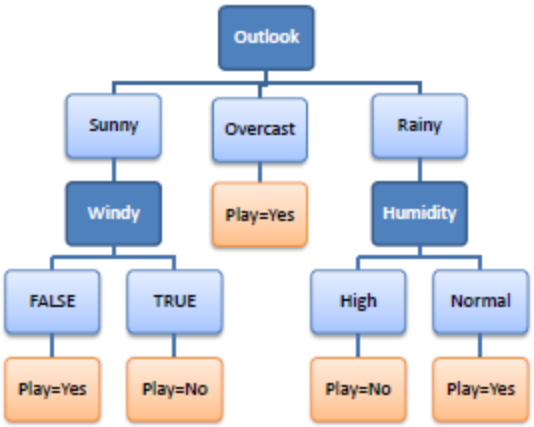
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



Attribute selection criteria

Information gain can be based on…

- Entropy
  - As seen in the previous chapter
- Gini impurity
  -
- Misclassification error

$$H(X) = -\sum_{x \in V(X)} p(x) \log p(x) = \sum_{x \in V(X)} p(x) \log \frac{1}{p(x)}$$

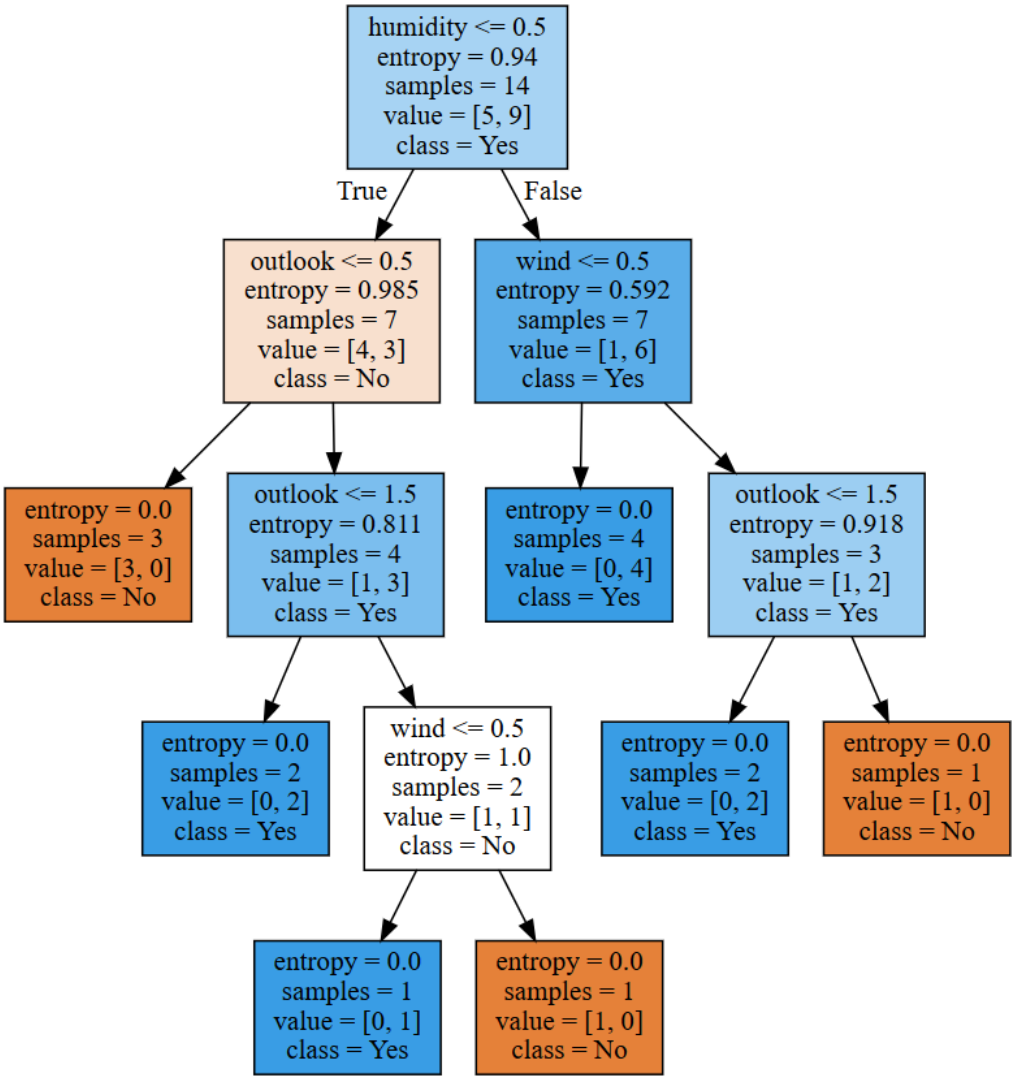
$$H(X) = \sum_{x \in V(X)} p(x)\,(1 - p(x))$$

$$H(X) = 1 - \max_{x \in V(X)} p(x)$$

How to handle continuous attributes?

CART - Classification And Regression Tree - can be used to handle continuous attributes.

- Binary branching
- **Binning:** split on the most discriminative threshold



7. Bayesian learning

Introduction

Each observed training example decreases or increases the estimated probability that a hypothesis is correct.

- Prior knowledge and observed data are used to estimate the probability of a hypothesis.
- Accommodates hypotheses that make probabilistic predictions

Bayes' theorem

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$$

Where:

- h: hypothesis
- D: training data
- P(h): prior probability that hypothesis h holds
- P(D): prior probability that training data will be observed
- P(h|D): posterior probability of h (given observed data)
- P(D|h): posterior probability of D given hypothesis h

Maximum a posteriori

$$h_{map} = \underset{h \in H}{argmax}\, P(h|D)$$

$$= \underset{h \in H}{argmax} \frac{p(D|h) \cdot P(h)}{P(D)}$$

$$= \underset{h \in H}{argmax}\, P(D|h) \cdot P(h)$$

$$= \underset{h \in H}{argmax}\, P(D|h) = h_{ML}$$

Where:

- HMAP: maximum a posteriori hypothesis
- ML: maximum likelihood hypothesis

Bayesian inference

Uses probabilities/likelihood that samples of a certain class have particular properties, in order to calculate the most probable class for a (new) sample.

Naive assumption: Attribute values are conditionally independent, given the class!

Classifier

Naive bayes classifier
Suitable for a task where samples are described as a conjunction of attribute values and a target function that takes values from a finite set C.

- **Probabilistic classifier**
  - Tells a probability of class membership for each class.
- **Generative classifier**
  - Builds a model of how a class could generate some input data.
  -

$$c_{NB} = \underset{c \in C}{argmax}\ P(c) \cdot \prod_{i=1}^{n} P(x_i|c)$$

Let's apply the naive bayes classifier to the weather data from the lecture slides!

Frequency table		**Play Golf**	
		Yes	**No**
Outlook	**Sunny**	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood table		**Play Golf**	
		Yes	**No**
Outlook	**Sunny**	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5
		9/14	**5/14**

To clarify:

- P(Sunny|Yes): 3/9
- P(Yes): 9/14

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	**3**	**2**
	Overcast	**4**	**0**
	Rainy	**2**	**3**

		Play Golf	
		Yes	No
Humidity	High	**3**	**4**
	Normal	**6**	**1**

		Play Golf	
		Yes	No
Temp.	Hot	**2**	**2**
	Mild	**4**	**2**
	Cool	**3**	**1**

		Play Golf	
		Yes	No
Windy	False	**6**	**2**
	True	**3**	**3**

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	**3/9**	**2/5**
	Overcast	**4/9**	**0/5**
	Rainy	**2/9**	**3/5**

		Play Golf	
		Yes	No
Humidity	High	**3/9**	**4/5**
	Normal	**6/9**	**1/5**

		Play Golf	
		Yes	No
Temp.	Hot	**2/9**	**2/5**
	Mild	**4/9**	**2/5**
	Cool	**3/9**	**1/5**

		Play Golf	
		Yes	No
Windy	False	**6/9**	**2/5**
	True	**3/9**	**3/5**

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \times \prod_{i=1}^{4} P(x_i|c)$$

$$c_{NB} = \operatorname{argmax} \begin{cases} P(Yes) \times \prod_{i=1}^{4} P(x_i|Yes) = \\ = \frac{9}{14} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \approx 0.0079 \\ P(No) \times \prod_{i=1}^{4} P(x_i|No) = \\ = \frac{5}{14} \times \frac{2}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \approx 0.0137 \end{cases}$$

How do we deal with zero probabilities?
Smoothing! Why handle them?

- Can be as simple as a +1
  - Also known as Laplace smoothing
- More general smoothing
  - Value between 0 and 1

$$P(No) \times \prod_{i=1}^{4} P(x_i|No) =$$

$$= \frac{5}{14} \times \frac{0}{5} \times \cdots = 0$$

$$c_{NB} = \text{argmax} \begin{cases} P(Yes) \times \prod_{i=1}^{4} P(x_i|Yes) = \\ = \frac{\mathbf{9}}{\mathbf{14}} \times \frac{5}{12} \times \frac{4}{12} \times \frac{4}{11} \times \frac{4}{11} = e_{yes} \\ P(No) \times \prod_{i=1}^{4} P(x_i|No) = \\ = \frac{\mathbf{5}}{\mathbf{14}} \times \frac{\mathbf{1}}{\mathbf{8}} \times \frac{2}{8} \times \frac{5}{7} \times \frac{4}{7} = e_{no} \end{cases}$$

Use cases

How can we use a Naive Bayes classifier?

- Spam detection
- Sentiment analysis
- Language identification
- Authorship attribution
- Topic/genre classification

Multinomial and Multivariate Naive Bayes

Multinomial Naive Bayes
Naive bayes variant, working with a multinomial distribution of data.

- Feature number is changing across samples
  - Features are token positions
- Each feature can take as many values as the size of the vocabulary
- Unseen words in the training set should be discarded

- Does not model the absence of a keyword

Multivariate Naive Bayes
Naive bayes variant with fixed feature numbers
- Used when data is discrete
- Each feature can take a 0 or 1 value
    - Japan = 1? Japan occurs in a document
    - Tokyo = 0? Tokyo does not occur in a document
- Unseen words in the training set are discarded
- Does model the absence of a keyword

Random forest

What is a random forest?
- Ensemble of decision trees
- Individual trees are not great…
    - But together they perform well!
- More immune to noise than decision trees

How does it work?
1. Set a number of trees in the forest: N
2. (Randomly) sample a part K_i of the training data D: $K_i \subset D$
3. Train a decision tree T_i on K_i
4. Given an unseen instance u...
   $Forest(u) = most\_common(\{ T_i(u) \mid i=1...N \})$

8. Evaluation and revision

Introduction

Why evaluate?
It provides testing conditions that convincingly suggest that your algorithm will perform well on other people's data, out in the real world

Accuracy is not good enough
Accuracy as a measure is not informative enough…
- One number cannot tell what the shortcomings of a system could be
- It does not take imbalanced class distribution into account

Confusion matrix

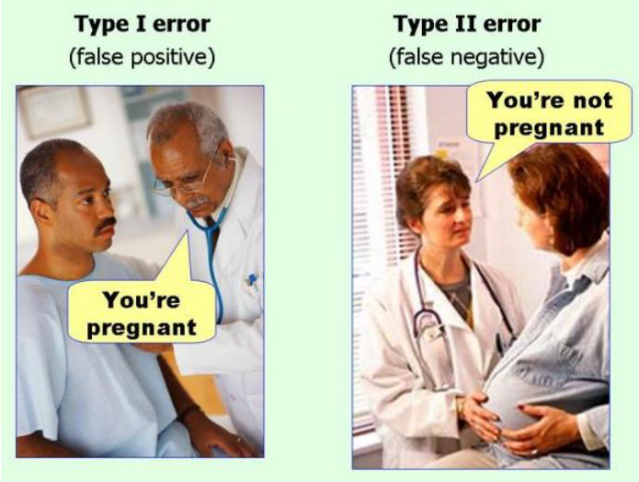
Also known as a **contingency matrix**.

	Prediction		
Gold standard		**Positive**	**Negative**
	Positive	65	7
	Negative	13	25

	Prediction		
Gold standard		**Positive**	**Negative**
	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Where:

- FP: Type I error
- FN: Type II error



Single-number metrics

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

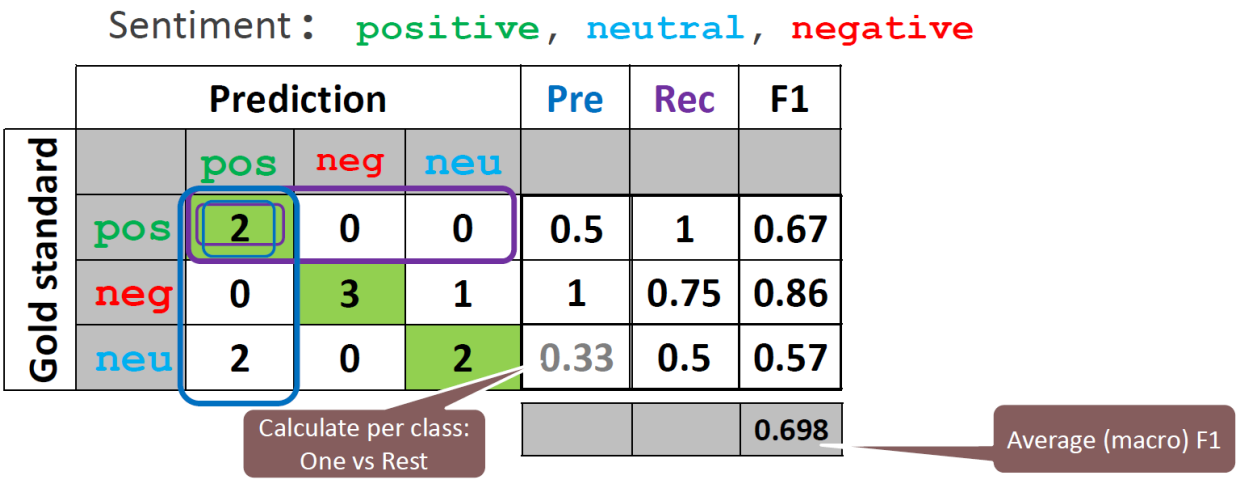
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

A variant of the F1-score is the F_β-score.

$$F_\beta = \left(1 + \beta^2\right) \frac{PR}{\beta^2 P + R}$$

$$F_1 = (1 + 1) \frac{PR}{1^2 P + R}$$

F1-score for multiple classes



Youden's J statistic

Uses all numbers from the confusion matrix.

$$J = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1 = TPR - FPR$$

$$True\ positive\ rate = TPR = \frac{TP}{TP + FN} \qquad True\ negative\ rate = TNR = \frac{TN}{TN + FP}$$

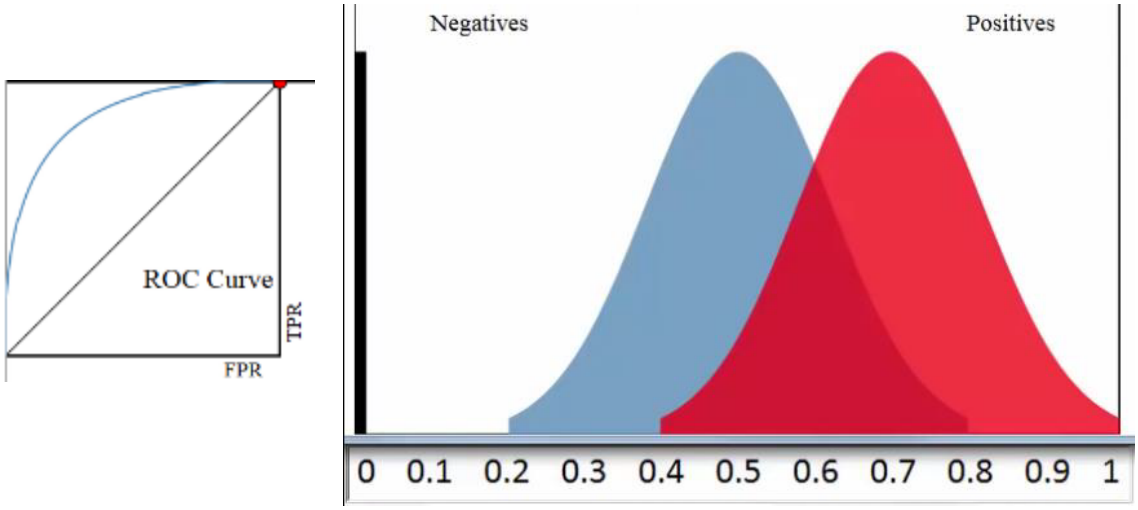
$$False\ positive\ rate = FPR = 1 - TNR = 1 - \frac{TN}{TN + FP}$$

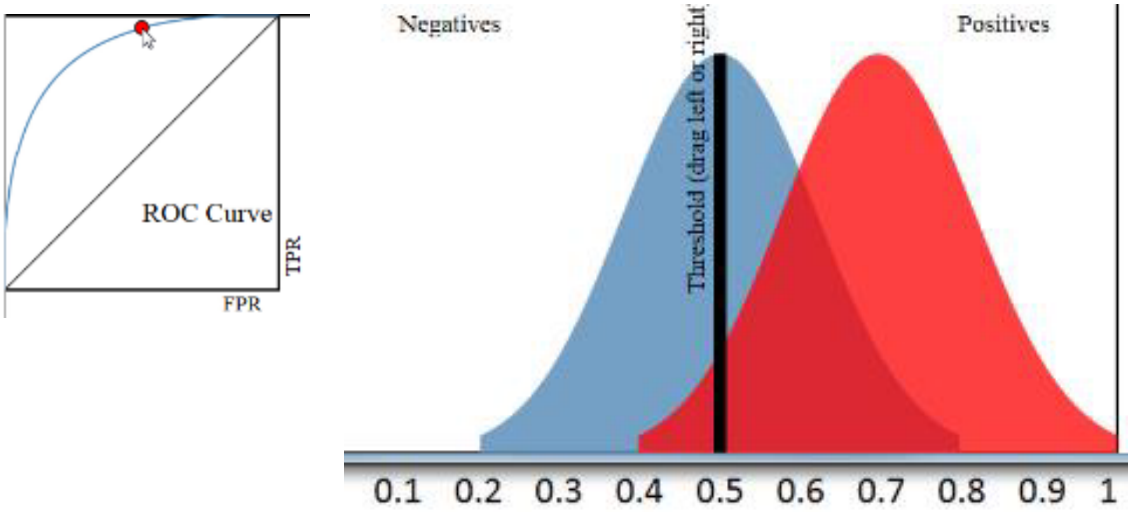
Curves

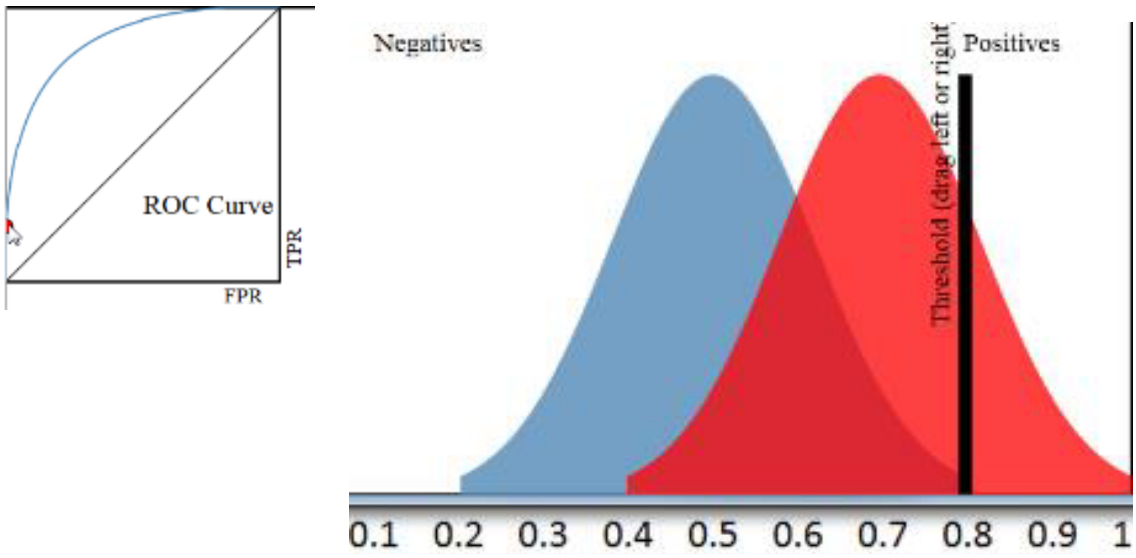
What is the ROC curve?

ROC stands for Receiver Operating Characteristics.

- Takes into account the distributions assigned by the classifier to positive and negative samples
- In short:
  - FPR is bad
  - TPR is good
- A threshold value corresponds to a point of the ROC curve


ROC Curve
TPR
FPR
Negatives
Positives
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

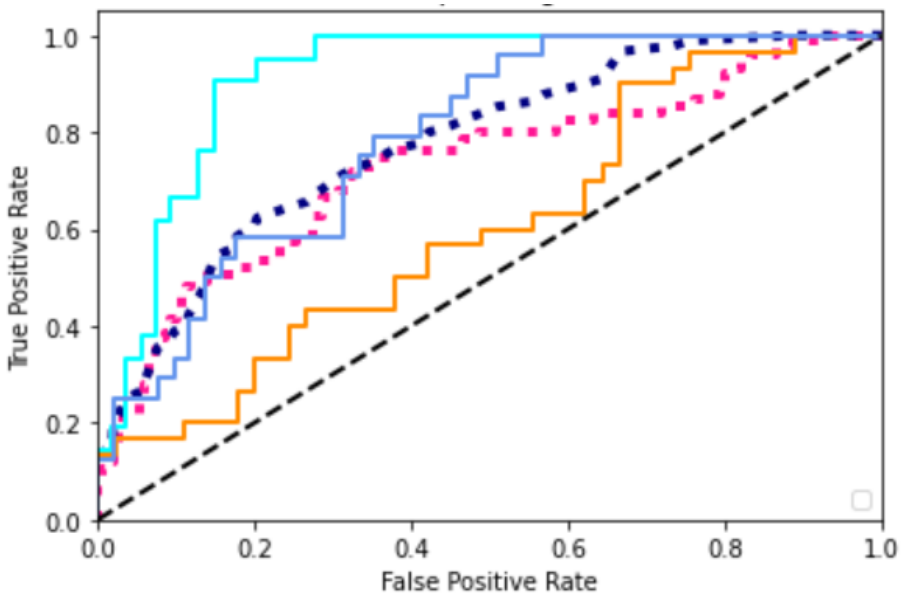

ROC Curve
TPR
FPR
Negatives
Threshold (drag left or right)
Positives
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


ROC Curve
TPR
FPR
Negatives
Threshold (drag left or right)
Positives
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

What is the AUC curve?

AUC stands for Area Under the ROC Curve. It works internally the same as ROC.

- Abstract over the thresholds by comparing ROC curves
    - Compares the areas under the curves



Small data challenges

Small data might not be suitable for splitting into training, test and/or development…

K-fold cross validation

K disjoint equal parts

- K = |D|
    - Leave-one-out cross-validation
- Stratified CV: keep class distribution per part
- Repeated CV: repeat K-fold CV N times

Bootstrapping

- Size M and repeat N times
- Randomly pick m training samples
- Test set is the rest of the samples
- N=20 or 30; m = |D|
    - |D|: size of the total data

Overfitting

Too many features in the model, with respect to the data size.

- Features could be disproportionately distributed
- Too much info in the annotation?

- Check the representativeness and balance of the corpus
- In short:
    - **When a feature boosts the evaluation score on training and development set, but lowers on the final test set**.

Revising the work

Revisions can be made in or for…

- The model
- The annotation specifications
- The annotation task

Why revise?

- The corpus might not be balanced anymore
- The corpus might not be representative anymore, with respect to the original task
    - A trained model could still be used on a new, related corpus

Keep in mind…

- Do all tags and attributed form distinct categories?
    - Merge infrequent tags with other tags
- Do not change guidelines if you plan to release the corpus!
- Reflect on the annotation tools and the annotators
- Examine more linguistic aspects of the corpus for training

Reporting about the work

In general, people are interested to hear the details of corpus creation

- Annotators
- Corpus explorers
- Language engineers
- Guideline explorers

Share the corpus!

- Check for potential copyright issues
- Check for changes to the original corpus
- Provide code, if any

Explain the model

- Which linguistic theories were used as the foundation of the work?
- Make your reasoning behind the specification clear

Explain the annotation task itself

- How many annotators participated in the task?

- How proficient were the annotators?
    - Did they receive training?
- The purpose, type and computation of the IAA
- Annotation and adjudication tools used
- Assembly of the gold standard
- Information about the adjudicators
    - Why are they the ones adjudicating?
    - How do they reach consensus?
- Aspects that affected agreement scores and annotation quality
- Major sources of confusion/disagreement among annotators and adjudicators

END

This summary was based on the lecture slides by Lasha Abzianidze.