Outline

- Motivation examples
- Supervised Machine Learning
- Searching for a good hypothesis
- Development cycle
- Brief overview of the course
- Examination requirements
Motivation example

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.
- ...

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
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.

The company has just launched a new line of small, low-priced computers.

Draw a line that passes through the points P and Q.

This has been a very popular new line.
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 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!
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.

<table>
<thead>
<tr>
<th>sentence</th>
<th>sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>I've got Inspector Jackson on the line for you.</td>
<td>PHONE</td>
</tr>
<tr>
<td>Outside, a line of customers waited to get in.</td>
<td>FORMATION</td>
</tr>
<tr>
<td>These companies rent private telephone lines.</td>
<td>PHONE</td>
</tr>
<tr>
<td>Please hold the line.</td>
<td>PHONE</td>
</tr>
<tr>
<td>He quoted a few lines from Shakespeare.</td>
<td>TEXT</td>
</tr>
<tr>
<td>He drew a line on the chart.</td>
<td>DIVISION</td>
</tr>
<tr>
<td>She hung the washing on the line.</td>
<td>CORD</td>
</tr>
</tbody>
</table>
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.**
To choose an effective set of features we always need our intuition. Only then all experiments with data can start.

A few example hints:

<table>
<thead>
<tr>
<th>class</th>
<th>a feature to recognize the class – will be useful?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD</td>
<td>immediately preceding word</td>
</tr>
<tr>
<td>FORMATION</td>
<td>immediately following word</td>
</tr>
<tr>
<td>PHONE</td>
<td>can be often recognized by characteristic verbs</td>
</tr>
</tbody>
</table>
“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

\[ \text{example} = \text{feature vector} + \text{output value} \]
Data instances

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

\[
\text{data instance} = \text{feature vector} \ (\text{+ output value, if it is known})
\]

A data instance is either a feature vector or a complete example.
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 $E$. 
**Supervised learning process**

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

- **Training examples**: data instances in the form of feature vectors and correct target values
- **Machine learning method + parameters**: trained model should represent the learned knowledge
- **Predictor**:
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
Idealized model of supervised learning

- $x_i$ are feature vectors, $y_i$ are true predictions
- prediction function $h^*$ is the “best” of all possible hypotheses $h$
- learning process is searching for $h^*$, which means to search the hypothesis space and minimize a predefined loss function
- ideally, the learning process results in $h^*$ so that predicted $\hat{y}_i = h^*(x_i)$ is equal to the true target values $y_i$
A loss function $L(\hat{y}, y)$ measures the cost of predicting $\hat{y}$ when the true value is $y$. Commonly used loss functions are

- **Squared loss** $L(\hat{y}, y) = (\hat{y} - y)^2$
  
  for regression

- **Zero-one loss** $L(\hat{y}, y) = I(\hat{y} \neq y)$
  
  for classification; *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 $L(\hat{y}, y)$ 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. $\sum L(\hat{y}, y)$ or $E[L(\hat{y}, y)]$.

- 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
• How different people call values that describe objects

<table>
<thead>
<tr>
<th></th>
<th>observed (known) object characteristics</th>
<th>values or categories to be predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>computer scientists</strong></td>
<td>features</td>
<td>(target) value or class</td>
</tr>
<tr>
<td><strong>mathematicians</strong></td>
<td>attributes or predictors</td>
<td>response (value) or output value</td>
</tr>
</tbody>
</table>

NPFL054, 2016 Hladká & Holub Lecture 1, page 20/39
Data preprocessing and feature extraction

- Data preprocessing
  - Original objects
  - Analyzed objects
  - Feature vectors

- Primary feature extraction
Feature extraction and feature selection

Development data → primary feature extraction → Initial feature vectors

Initial feature vectors → advanced feature extraction → Transformed feature vectors

Initial feature vectors → feature selection → Reduced feature vectors
Sample error and generalization error

**Sample error** of a hypothesis $h$ with respect to a data sample $S$ of the size $n$ is usually measured as follows

- for **regression**: **mean squared error** $\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$

- for **classification**: **classification error** $\frac{1}{n} \sum_{i=1}^{n} I(\hat{y}_i \neq y_i)$

**Generalization error** (aka “true error” or “expected error”) measures how well a hypothesis $h$ generalizes beyond the used training data set, to unseen data with distribution $D$. Usually it is defined as follows

- for **regression**: $\text{error}_D(h) = E (\hat{y}_i - y_i)^2$
- for **classification**: $\text{error}_D(h) = \Pr (\hat{y}_i \neq y_i)$
Accuracy and error rate

To measure the performance of classification tasks we often use (sample) **accuracy** and (sample) **error rate**

**Sample accuracy** is the number of correctly predicted examples divided by the number of all examples in the predicted set.

**Sample error rate** is equal to $1 - \text{accuracy}$.

**Training error rate** is the sample error rate measured on the training data set.

**Test error rate** is the sample error rate measured on the test data set.
Machine learning process — development cycle

Formulating the task

Getting data → Building classifier → Final evaluation

ML method selection

Classifier to use

Development cycle

Feature engineering
Classifier training
Development evaluation
Parameter tuning
Terminological note on building predictors

The purpose of the learning process is search for the best prediction function parameters

<table>
<thead>
<tr>
<th>learning parameters</th>
<th>hypothesis parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>= parameters of the learning process</td>
<td>= parameters of the prediction function</td>
</tr>
</tbody>
</table>

- **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).
All subsets should be selected randomly in order to represent the characteristic distribution of both feature values and target values in the available set of examples.
Evaluation – basic scheme

- Test data
- True classes
- Classifier
- Prediction
- Comparison
- Evaluation
Minimizing generalization error

Finding a model that minimizes generalization error... is one of central goals of the machine learning process.
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: $Y = SENSE$
   $SENSE = \{CORD, DIVISION, FORMATION, PHONE, PRODUCT, TEXT\}$
Step 1: Getting feature vectors

Original objects → Analyzed objects → Feature vectors

Data preprocessing → Primary feature extraction
Getting data

**Step 1: Getting feature vectors**

- Features as variables $A_1, \ldots, A_m$
  - **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 $x_1, \ldots, x_m$, $x_i \in A_i$
- Each object represented as feature vector $\mathbf{x} = \langle x_1, \ldots, x_m \rangle$
- Feature vectors are elements in an $m$-dimensional feature space
- Set of instances $X = \{ \mathbf{x} : \mathbf{x} = \langle x_1, \ldots, x_m \rangle, x_i \in A_i \}$. 

Getting data

Step 1: Getting feature vectors – Example

Outside, a line of customers waited to get in. Please hold the line. She hung the washing on the line. He drew a line on the chart. …

1. preprocessing
2. feature extraction
### Example feature vectors – the WSD task

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>safety</td>
<td>special</td>
<td>install</td>
<td>inside</td>
<td>NN</td>
<td>IN</td>
<td>DT</td>
<td>lines</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>class</td>
<td>across</td>
<td>reach</td>
<td>.</td>
<td>NN</td>
<td>.</td>
<td>X</td>
<td>lines</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>fine</td>
<td>the</td>
<td>walk</td>
<td>between</td>
<td>JJ</td>
<td>IN</td>
<td>JJ</td>
<td>line</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>fine</td>
<td>``</td>
<td>a</td>
<td>between</td>
<td>JJ</td>
<td>IN</td>
<td>VBG</td>
<td>line</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>a</td>
<td>draw</td>
<td>to</td>
<td>between</td>
<td>DT</td>
<td>IN</td>
<td>NNS</td>
<td>line</td>
<td>dobj</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>a</td>
<td>draw</td>
<td>to</td>
<td>between</td>
<td>DT</td>
<td>IN</td>
<td>NNS</td>
<td>line</td>
<td>dobj</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>when</td>
<td>,</td>
<td>of</td>
<td>JJ</td>
<td>IN</td>
<td>NNS</td>
<td>lines</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>in</td>
<td>patiently</td>
<td>to</td>
<td>JJ</td>
<td>TO</td>
<td>VB</td>
<td>lines</td>
<td>prep_in</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>the</td>
<td>but</td>
<td>delay</td>
<td>JJ</td>
<td>VBD</td>
<td>DT</td>
<td>lines</td>
<td>nsubj</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>car</td>
<td>the</td>
<td>X</td>
<td>affect</td>
<td>NN</td>
<td>VBN</td>
<td>IN</td>
<td>lines</td>
<td>nsubj</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>establish</td>
<td>of</td>
<td>marketing</td>
<td>such</td>
<td>VBN</td>
<td>JJ</td>
<td>IN</td>
<td>lines</td>
<td>prep_of</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>main</td>
<td>few</td>
<td>a</td>
<td>and</td>
<td>JJ</td>
<td>CC</td>
<td>RB</td>
<td>lines</td>
<td>prep_on</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>computer</td>
<td>new</td>
<td>the</td>
<td>to</td>
<td>NN</td>
<td>TO</td>
<td>VB</td>
<td>line</td>
<td>dobj</td>
</tr>
</tbody>
</table>

Step 2: Assigning true prediction

- 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 $Data = \{\langle x, y \rangle : x \in X, y \in Y\}$. 
Step 2: Assigning true prediction

Example: $Y = \text{SENSE} = \{\text{CORD, DIVISION, FORMATION, PHONE, PRODUCT, TEXT}\}$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cord</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>safety</td>
<td>special</td>
<td>install</td>
<td>inside</td>
<td>NN</td>
<td>IN</td>
<td>DT</td>
<td>lines</td>
<td>dobj</td>
</tr>
<tr>
<td>division</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>class</td>
<td>across</td>
<td>reach</td>
<td>.</td>
<td>NN</td>
<td>.</td>
<td>X</td>
<td>lines</td>
<td>prep_across</td>
</tr>
<tr>
<td>division</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>fine</td>
<td>the</td>
<td>walk</td>
<td>between</td>
<td>JJ</td>
<td>IN</td>
<td>JJ</td>
<td>line</td>
<td>dobj</td>
</tr>
<tr>
<td>division</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>fine</td>
<td>a</td>
<td>between</td>
<td>JJ</td>
<td>IN</td>
<td>VBG</td>
<td>line</td>
<td>dobj</td>
<td></td>
</tr>
<tr>
<td>division</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>a</td>
<td>draw</td>
<td>to</td>
<td>between</td>
<td>DT</td>
<td>IN</td>
<td>NNS</td>
<td>line</td>
<td>dobj</td>
</tr>
<tr>
<td>division</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>a</td>
<td>draw</td>
<td>to</td>
<td>between</td>
<td>DT</td>
<td>IN</td>
<td>NNS</td>
<td>line</td>
<td>dobj</td>
</tr>
<tr>
<td>formation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>when</td>
<td>,</td>
<td>of</td>
<td>JJ</td>
<td>IN</td>
<td>NNS</td>
<td>lines</td>
<td>nsubj</td>
</tr>
<tr>
<td>formation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>in</td>
<td>patiently</td>
<td>to</td>
<td>JJ</td>
<td>TO</td>
<td>VB</td>
<td>lines</td>
<td>prep_in</td>
</tr>
<tr>
<td>formation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>long</td>
<td>the</td>
<td>but</td>
<td>delay</td>
<td>JJ</td>
<td>VBD</td>
<td>DT</td>
<td>lines</td>
<td>nsubj</td>
</tr>
<tr>
<td>product</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>car</td>
<td>the</td>
<td>X</td>
<td>affect</td>
<td>NN</td>
<td>VBN</td>
<td>IN</td>
<td>lines</td>
<td>nsubj</td>
</tr>
<tr>
<td>product</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>establish</td>
<td>of</td>
<td>marketing</td>
<td>such</td>
<td>VBN</td>
<td>JJ</td>
<td>IN</td>
<td>lines</td>
<td>prep_of</td>
</tr>
<tr>
<td>product</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>main</td>
<td>few</td>
<td>a</td>
<td>and</td>
<td>JJ</td>
<td>CC</td>
<td>RB</td>
<td>lines</td>
<td>prep_on</td>
</tr>
<tr>
<td>product</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>computer</td>
<td>new</td>
<td>the</td>
<td>to</td>
<td>NN</td>
<td>TO</td>
<td>VB</td>
<td>line</td>
<td>dobj</td>
</tr>
</tbody>
</table>
Step 2: Assigning true prediction

Example: $Y = \{red, blue\}$
Getting data

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

- $Train \subseteq Data$, $Test \subseteq Data$
- $Train \cap Test = \emptyset$
- $Train \cup Test = Data$
You should be familiar with the key machine learning terms

- Machine learning process
- Development cycle
- Examples, feature vector, data instance, gold standard data, training data, test data
- Manual annotation (true prediction)
- Model, hypothesis, predictor
- Supervised learning, unsupervised learning
- Classification, regression
- Overfitting