Artificial Intelligence in Automotive Technology

Johannes Betz / Prof. Dr.-Ing. Markus Lienkamp / Prof. Dr.-Ing. Boris Lohmann
<table>
<thead>
<tr>
<th>Lecture Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction: Artificial Intelligence</strong></td>
</tr>
<tr>
<td><strong>Practice 1</strong></td>
</tr>
<tr>
<td><strong>2 Perception</strong></td>
</tr>
<tr>
<td><strong>Practice 2</strong></td>
</tr>
<tr>
<td><strong>3 Supervised Learning: Regression</strong></td>
</tr>
<tr>
<td><strong>Practice 3</strong></td>
</tr>
<tr>
<td><strong>4 Supervised Learning: Classification</strong></td>
</tr>
<tr>
<td><strong>Practice 4</strong></td>
</tr>
<tr>
<td><strong>5 Unsupervised Learning: Clustering</strong></td>
</tr>
<tr>
<td>22.11.2018 – Jan Cedric Mertens</td>
</tr>
<tr>
<td><strong>Practice 5</strong></td>
</tr>
<tr>
<td>22.11.2018 – Jan Cedric Mertens</td>
</tr>
</tbody>
</table>
Feedback from last week

- Specific feedback for the lecture content: Thanks for the feedback, we will integrate all your input for this lecture for the next year

- Practice Session:
  - Jupyter Notebook seems to be a good solution for coding tasks
  - Why Coding? Helps to understand the lecture objectives + gives you real life applications → Not only theoretical knowledge
  - More comments in the code
  - Coding together → Is not a feasible solution:
    - Different coding knowledge
    - Different coding speed

- Homework Feedback:
  - From now on: Feedback for the right solution only after the quiz is closed
Objectives for Lecture 4: Classification

After the lecture you are able to…

… understand the concept of classification, its association to pattern recognition and the urge for machine learning.

… acquire labeled training data and prepare it for the training and validation phase.

… plan the basic workflow for an arbitrary supervised learning problem.

… understand the concepts of different classification methods together with their pros and cons.

… implement, train and use a classification method with Python libraries.

… understand how classification can be used in the perception for automated vehicles.

… analyze the quality of a given classifier regarding to different criteria.
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation

2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine

3. Chapter: Application

4. Summary
Classification

“Systematic arrangement in groups or categories according to established criteria” [13]
Classification

“Systematic arrangement in groups or categories according to established criteria” [13]
Classification

“Systematic arrangement in groups or categories according to established criteria” [13]
Method Overview

**Pattern Recognition**

- **Regression**
  - Predict *continuous* valued output
  - Supervised

- **Classification**
  - Predict *discrete* valued output
  - Supervised

- **Clustering**
  - Predict discrete valued output
  - Unsupervised
Method Overview

- Regression
  - House pricing
  - Number of sales
  - Persons weight

- Classification
  - Object detection
  - Spam detection
  - Cancer detection

- Clustering
  - Genome patterns
  - Google news
  - Pointcloud (Lidar) processing
General Approach

Data (Features)

Classifier

Classes

Email (Keywords, …)  Spam?  Yes/No
Tumor (Size, …)  Malignant?  Yes/No
Object (Color, …)  What type?  Cat/Car/Fruit/…
Classic Method vs. Machine Learning Method

- **Classic Method**
  - E.g. Decision tree
    - Use a-priori knowledge to formulate classification rules

- **Advantages of machine learning**
  - Automatic generation of a-priori knowledge
  - Automatic generation of complex classification rules
  - Suitable for extreme large datasets
Classification - Example

Object Classification → Object Detection → Object Tracking
Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation

2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine

3. Chapter: Application

4. Summary
Formal Definition - Classification

\[ C_M(\theta): D \rightarrow Y \]

- Classifier \( C \)
- Model \( M \) with parameter \( \theta \)
- Dataspace \( D \)
- Labels \( Y \)
- Training Data \( O \subseteq D \) with known labels

- Training: Given \( O \), find optimal parameter \( \theta \)
- Classification: Apply \( C_M(\theta) \) on objects from \( D \)
Supervised Learning - Classification

Labeled Data → Training-Set → Classifier → Adjustment

Training
Classifier Training
Supervised Learning - Classification

- Labeled Data
- Training-Set
- Classifier
- Test-Set
- (Hidden labels)
- Adjustment
- Quality
- Training
- Validation
Quality Measures for Classifiers

Scalability
Compactness
Accuracy
Interpretability
Efficiency
Robustness
Quality Measures for Classifiers

- Classification accuracy or classification error (complementary)
  - Loss Functions
- Compactness of the model
  - decision tree size; number of decision rules
- Interpretability of the model
  - Insights and understanding of the data provided by the model
- Efficiency
  - Time to generate the model (training time)
  - Time to apply the model (prediction time)
- Scalability for large databases
  - Efficiency in disk-resident databases
- Robustness
  - Robust against noise or missing values
Evaluation of Classifiers

- **k-fold Cross Validation**
  - Decompose data set evenly into k subsets of (nearly) equal size
  - Iteratively use k–1 partitions as training data and the remaining single partition as test data.

- **Additional requirement: stratified folds**
  - Class distributions in training and test set should represent the class distribution in D (or at least in O)

- **Standard: 10-fold stratified cross validation**
## Confusion Matrix

<table>
<thead>
<tr>
<th>Correct Label</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>45</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Class 2</td>
<td>3</td>
<td>44</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Class 3</td>
<td>0</td>
<td>0</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>Class 4</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>37</td>
</tr>
</tbody>
</table>

- **Recall:** $\frac{TP}{TP+FN}$
- **Precision:** $\frac{TP}{TP+FP}$
- **Specificity:** $\frac{TN}{TN+FP}$

**Classified as**

- True Positives: TP
- False Positives: FP
- True Negatives: TN
- False Negatives: FN
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation

2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine

3. Chapter: Application

4. Summary
Methods

- Decision Trees
- Logistic Regression
- Nearest Neighbors
- Support Vector Machine
- Neural Networks
- …
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation

2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine

3. Chapter: Application

4. Summary
Recap Linear Regression

\[ y = h_\theta(x), \quad y \in \mathbb{R} \]
Linear Regression for Classification

\[ y = h_\theta(x), \quad y \in \mathbb{R} \]
Sigmoid Function

\[ g(z) = \frac{1}{1 + e^{-z}} \]
Logistic Regression

Probabilistic classification:

\[ y = g_\theta(h_\theta(x)) \quad y \in ]0,1[ \]
Discussion Logistic Regression

- **Pro:**
  - **Implementation:** Easy to use
  - **Probabilistic:** Probability of an object being in a certain class
  - **Computation:** Quick training phase
  - **Insights:** Produces understandable models

- **Contra:**
  - **Linearity:** Hard to adopt to non linear problems
  - **Overfitting:** Training data has to be well chosen
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation
2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine
3. Chapter: Application
4. Summary
Nearest Neighbor

Classify a new object based on its nearest neighbor

Size

Weight
Nearest Neighbor - Instance based learning

- No training and test phase
  - No generated model
- Store labeled training data
  - Points in a metric space
- Process training data when a new object should be classified
  - „lazy evaluation“
- Tradeoff between time and complexity
  - Hard to build a model based on a large dataset but it is easy to use
  - Easy to just save the large dataset but hard to search
Nearest Neighbor Variants

- **NN Classifier**
  - Consider only the nearest neighbor

- **k-NN Classifier**
  - Consider k nearest neighbors ($k>1$)

- **Weighted k-NN Classifier**
  - Consider the weighted distances to the k nearest neighbors

- **Mean-based NN Classifier**
  - Consider the closest mean position of a class
Nearest Neighbor Variants

- NN Classifier
  - Consider only the nearest neighbor
Nearest Neighbor Variants

- k-NN Classifier
  - Consider k nearest neighbors (k>1)
Nearest Neighbor Variants

- Weighted k-NN Classifier
  - Use weights for the classes of the K nearest neighbors
Nearest Neighbor Variants

- Mean-based NN Classifier
  - Consider the closest mean position of a class
k-NN Classifier

- How to choose k?
  - Generalization vs Overfitting
  - Large k: Many objects from different classes
  - Small k: Sensitivity against outliers
  - Practice: $1 \ll k < 10$
Weighted k-NN Classifier

- How to weight the neighbors?
  - Frequency of the neighbors class
    - \( w_i = \frac{1}{\text{frequency}_i} \)
  - Distance to the neighbor
    - \( w_i = \frac{1}{\text{distance}_i^2} \)

k=7

Normal  ●
Weighted (Frequency)  ●
Weighted (Distance)  ●
Discussion NN Classifier

- **Pro:**
  - **Applicability:** Easy to calculate distances
  - **Accuracy:** Great results for many applications
  - **Incremental:** Easy adoption of new training data
  - **Robust:** Scopes with noise by averaging (k-NN)

- **Contra:**
  - **Efficiency:** Processing grows with training data $\mathcal{O}(n)$
    - Can be reduced to $\mathcal{O}(\log n)$ with an index structure (requires training phase)
  - **Dimensionality:** Not every dimension is relevant
    - Weight dimensions (scale axes)

- **Neutral**
  - Does not produce explicit knowledge about classes
Supervised Learning: Classification  
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann  
(Jan Cedric Mertens, M. Sc.)  

Agenda

1. Chapter: Introduction  
   1.1 Overview  
   1.2 Training and Validation  

2. Chapter: Methods  
   2.1 Logistic Regression  
   2.2 Nearest Neighbors  
   **2.3 Support Vector Machine**  

3. Chapter: Application  

4. Summary
Support Vector Machine (SVM)

- Linear separation
  - Objects in $\mathbb{R}^d$
  - Two classes
  - Hyperplane separates both classes

- Training
  - Compute Hyperplane

- Classification
  - Distance to Hyperplane
SVM - Maximum Margin Hyperplane (MMH)

- Max. distance to Hyperplane
  - At least $\delta$ (Margin)

- High generalization
  - Maximal stable

- Support Vector
  - Only depends on objects with distance $\delta$
SVM – Formal Definition

- **Training data:** \((x_1, y_1) \ldots (x_n, y_n)\)
  with \(x \in \mathbb{R}^d, y \in \{-1,1\}\)

- **Hyperplane:** \(w \cdot x - b = 0\)
  with \(w\) normal vector, \(\frac{b}{\|w\|}\) offset from origin,

- **Margin:** \(\delta = \frac{1}{\|w\|}\)

- **Training:** Minimize \(\|w\|\)
  with \(y_i(w \cdot x_i - b) \geq 1\) for \(i = 1 \ldots n\)

- **Classification:**
  if \((w \cdot x - b) \geq 0\), \(y = 1\); else \(y = -1\)
  with Data \(x \in \mathbb{R}^d\)
SVM - Soft Margin

- Linear separation
  - Not always possible
  - Not always optimal

- Tradeoff between Error and Margin
  - Allow classification error to maximize margin
SVM - Space Transformation

- Non Linear data
  - Too many errors with Soft Margin

- Use higher dimensional space
  - Increase dimensions until linear separation is possible
  - Transform Hyperplane back to lower dimensions
  - Hyperplane becomes non-linear

- Example: Quadratic transformation
  - Hyperplane becomes polynomial of degree 2
SVM - Kernel Machines Visualisation
SVM - Kernel Machines

- Space transformations
  - Lower to higher dimensions
  - Computational complex

- Hyperplane transformation
  - Higher to lower dimension
  - Feasibility not guaranteed
  - Computational complex

- Kernel
  - Computational elegant
  - Calculate dot product without full space transformation
SVM - Kernel Machines

- Replace the dot product with a non linear kernel function

- Polynomial:
  \[ k(x_i, x_j) = (x_i \cdot x_j)^d \]

- Gaussian radial bias function (RBF):
  \[ k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \text{ for } \gamma > 0 \]

- Linear, Sigmoid, Hyperbolic …
SVM - Kernel Example

- $f: \mathbb{R}^3 \rightarrow \mathbb{R}^9$
  
  $f(x) = (x_1 x_1, x_1 x_2, x_1 x_3, x_2 x_1, x_2 x_2, x_2 x_3, x_3 x_1, x_3 x_2, x_3 x_3)$
  
  $k(x, y) = (x \cdot y)^2$

- $x = (1,2,3), \ y = (4,5,6)$
  
  $f(x) = (1,2,3,2,4,6,3,6,9)$
  
  $f(y) = (16,20,24,20,25,30,24,30,36)$
  
  $f(x) \cdot f(y) = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$

- $k(x, y) = (4 + 10 + 18)^2 = 32^2 = 1024$

$\Rightarrow$ no transformation to $\mathbb{R}^9$ required
SVM - Kernel Machines

SVC with linear kernel

LinearSVC (linear kernel)

SVC with RBF kernel

SVC with polynomial (degree 3) kernel
Multi Class SVM

Combination of SVMs

1 vs. Rest

1 vs. 1
Discussion SVM

- **Pro:**
  - **Accuracy:** High classification rate
  - **Effective:** Even when number of dimensions > number of samples
  - **Robust:** Low tendency to overfitting
  - **Compact Models:** “Plane in Space”
  - **Versatile:** Different Kernel Function

- **Contra:**
  - **Efficiency:** Long training phase
  - **Complexity:** High implementation effort
  - **Black-Box:** Hard to interpret models
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation
2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine
3. Chapter: Application
4. Summary
Classification Problems

- Big Data
  - Find patterns
  - Make data usable
- Image classification
  - Handwritten Digits
  - X-Rays
- Music classification
  - Shazam
- Speech/Language classification
  - Siri/Alexa/Echo
- Fault detection
  - Quality control during production
Classification for automotive technology

- **Example: Perception**
  - Camera outputs pixel array
  - Classification adds value to each pixel
  - Pixel segmentation
  - Object detection
  - Object tracking
Vehicle Detection and Tracking
Vehicle Detection and Tracking

- Get training data
- Extract features from images
- Generate a model based on the features
- Take one video frame and classify the features of the sub-images
- Merge classified areas
Training Data

- Required label: „car“ or „no car“

- Required Images:
  - Same format used for classification
  - Representative for what we expect to find in the videostream
  - 8000 images (90 % training and 10 % test)
Training Data

- How to get labeled data?
  - Label data by yourself
  - Pay someone else to label your data
  - Let other label your data for free

- Collection of labeled data
  - Digits: MNIST
    - 70k images
  - Cars: KITTI
    - [www.cvlibs.net/datasets/kitti/](http://www.cvlibs.net/datasets/kitti/)
    - 80k images
Training Data

„It´s terrifying that both of these things are true at the same time in this world:

1. Computers drive cars around

2. The state of the art test to check that you´re not a computer is whether you can successful identify stop signs in pictures“

- Anonym
Feature Extraction

- Histogram of Oriented Gradients (HOG)
  - compressed & encoded version of the image
Build SVM Classifier

- Machine learning libraries (python)
  - scikit-learn (http://scikit-learn.org/)

```python
>>> from sklearn import svm
>>> clf = svm.SVC()
>>> clf.fit(training_features, training_labels)
>>> clf.score(test_features, test_labels)
>>> clf.predict(new_feature)
```

- Training: 1.44 Seconds
- Test: Accuracy = 0.9848
- Prediction: 0 or 1
Classify sub-images

- Produce sub-images of each frame for the classification
Merge classified areas

- Merge classes of sub-images
Final Output
Supervised Learning: Classification
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Jan Cedric Mertens, M. Sc.)

Agenda

1. Chapter: Introduction
   1.1 Overview
   1.2 Training and Validation
2. Chapter: Methods
   2.1 Logistic Regression
   2.2 Nearest Neighbors
   2.3 Support Vector Machine
3. Chapter: Application
4. Summary
Summary

What did we learn today:

- **Classification** is about assigning given classes to data.
- We need lots of **training data** to build a model for the classification.
- **Machine learning** can extract knowledge from huge datasets.
- Classification is a **supervised learning** problem.
- We need labeled data for training and validation (hidden label).
- We have several criteria to measure the **quality of a classifier**.
- The concepts of **Logistic regression**, **Nearest Neighbor** and **SVM**
- We can use linear regression together with a sigmoid function as classification method.
- Nearest Neighbor is an instance based learning method, no training is required.
Summary

What did we learn today:

- **SVMs** are linear classifier using a maximum margin hyperplane.
- With the **kernel trick**, SVMs can be used for non linear classification.
- Classification is very important for the **perception**, eg. in cars.
- Acquiring lots of **labeled data** is a problem.
- We have access to good and easy to use **python libraries** for classification.
- We have access to many **open source datasets** (eg. KITTI for car images).
- Training with big datasets can take a **long time**.
- We have to **partition, classify and then merge** images.
- We have to **extract features** from images for the classification.
Sources

- [2] https://funnyjunk.com/My+neighbours+like+this/funny-pictures/6231925/
- [4] https://www.youtube.com/watch?v=QopUtQobWJ0
- [5] https://www.youtube.com/watch?v=9NrALgHFwTo
- [12] https://samsclass.info/120/proj/captchas-021916.htm
Acknowledgment

- Machine Learning (Stanford/Coursera)
  - Andrew Ng
    https://www.coursera.org/learn/machine-learning

- Knowledge Discovery in Databases I (LMU)
  - Prof. Dr. Peer Kröger
    http://www.dbs.ifi.lmu.de/cms/studium_lehre/lehre_master/kdd1718/index.html