# Introduction to Machine Learning

Joschka Braun
Supervised Learning

November 3, 2020

# Contents

# 1 Machine Learning

## 1.1 Motivation

For many problems it is very difficult or almost impossible to the write all the possible explicit instructions necessary to return the correct output for a given input.

## 1.2 Definition

Machine learning describes the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data.

## 1.3 Advantages and Disadvantages

Advantages compared to conventional algorithms:

- solve previously unsolvable problems
- not every possible case and instruction has to be coded by hand

Disadvantages compared to conventional algorithms:

- reliant on good and large training data
- needs time, skill and a large amount computation to train model
- often it is difficult to understand how the algorithm is making its decisions, therefore the algorithm stays a black box

## 1.4 Examples

- Computer Vision
  - Medical diagnosis
  - Self driving cars
- Speech Recognition
- Online Fraud Detection
- Email Spam and Malware Filtering

## 1.5 History of Machine Learning

before 1950s:

New statistical methods are developed and improved

1950s:

Machine learning research with simple algorithms

1960s:

Bayesian methods are introduced for probabilistic inference in machine learning

1970s:

'AI Winter' caused by pessimism about machine learning effectiveness

1980s:

Rediscovery of back propagation causes a resurgence in machine learning research

1990s:

Work on machine learning shifts from a knowledge-driven approach to a data-driven approach

Support vector machines and recurrent neural networks become popular

2000s:

Support vector clustering and other kernel methods and unsupervised machine learning methods become widespread

2010s:

Deep learning becomes feasible, which leads to machine learning becoming integral to many widely used software services and applications

## 1.6 Machine Learning Algorithms playing Games

1997:

IBM's Deep Blue beats the world champion Garry Kasparov at chess

2011:

IBM's Watson beats two human champions in a Jeopardy! competition

2016:

Google's AlphaGo program beats Lee Sedol, the world champion at Go

# 2 Supervised Learning

When training data comprises examples of the input vectors $\mathbf{x} = (x_1, ..., x_N)^T$ along with their corresponding target vectors $\mathbf{t} = (t_1, ..., t_N)^T$ it is known as a supervised learning problem.

The goal of such supervised learning problems is to predict the target vector of new input vectors correctly.

Supervised learning problems can be grouped into regression and classification problems.

## 2.1 Comparison to Unsupervised Learning

When training data comprises only input vectors $\mathbf{x} = (x_1, ..., x_N)^T$ without corresponding target vectors it is known as a unsupervised learning problem.

The goal of such unsupervised learning problems may be to discover groups of similar examples within the training data, where it is called clustering, or to determine the distribution of the training data within the input space, known as density estimation, or to project the training data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

Unsupervised learning will be explained in Seminar 2 by Hannah Santvliet.

## 2.2 Comparison to Reinforcement Learning

In reinforcement learning problems there is no training data in the form of predefined input vectors. Typically there is a sequence of states and actions in which the learning algorithm interacts with its environment as an agent. Based on the its actions the agent is rewarded or punished.

The goal of such reinforcement learning problems is to find suitable actions to take in a given situation in order to maximize the reward and minimize the punishment.

## 2.3 Classification

The goal of classification problems is to assign each input vector one discrete output vectors called category. The number of possible outputs is therefore finite and the different outputs share no common metric.

Examples for classification problems:

- recognizing whether a given picture is a cat or a dog
- digit recognition
- spam detection

## 2.4 Regression

The goal of regression problems is to assign each input vector one or more continuous variables.

Examples of regression problems:

- estimating the age of a person from a picture
- predict the yield in a chemical manufacturing process in which the inputs consist of concentration of reactants, the temperature and the pressure
- evaluating the value of a chess position
- predicting housing prices from a portfolio

## 2.5 Bayes's Theorem

Bayes's theorem is stated mathematically as the following equation and expresses the conditional probability of an output Y given and input X.

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

for $P(X) \neq 0$

## 2.6 Generative Models

In general, a generative model explicitly models the actual distribution of each class. To get the conditional probability $P(Y \mid X)$, generative models estimate the prior $P(Y)$ and likelihood $P(X \mid Y)$ from training data and use Bayes rule to calculate the posterior $P(Y \mid X)$

Examples for generative models:

- Naive Bayes Classifier
- Bayesian Networks
- Hidden Markov models
- Gaussian mixture models

## 2.7 Discriminative Models

In general, a discriminative model models the decision boundary between the classes. To get the conditional probability $P(Y \mid X)$, discriminative models directly assume functional form for $P(Y \mid X)$ and estimate parameters of $P(Y \mid X)$ directly from training data.

Examples for discriminative models:

- Support Vector Machine
- Logistic Regression (Seminar 5 Mathias Neitzel)
- Conditional Random Fields
- Decision Trees
- Neural Networks (Seminar 6 Jan Feldman)

Discriminative Models often have higher accuracy than generative models, which mostly leads to better learning result. On the other hand discriminative models usually require multiple numerical optimization technique in comparison to generative models.

## 2.8 Ensemble learning

Uses multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

The ensemble consists of a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.

# 3 Examples of Supervised Learning

## 3.1 Naive Bayes Classifier

The Naive Bayes classifier, is a simple generative model used for classification problems, where the posterior probability $P(Y \mid X) = P(y)P(x_1 \mid y)P(x_2 \mid y, x_1)...P(x_n \mid y, x_1, ...x_{n-1})$ gets simplified. We see that the dependencies make it hard to infer $P(Y \mid X)$ as we need to condition the probability of $x_i$ on y and $x_1, x_2 \dots x_{i-1}$.

The Naive Bayes classifier, assumes that all the X are conditionally independent: $x_1 \perp\!\!\!\perp x_2 \perp\!\!\!\perp ... \perp\!\!\!\perp x_n \mid y$. With this assumption, now we can rewrite the posterior distribution as:

$$P(Y = 1 \mid X) = \frac{P(y = 1)P(x_1 \mid y = 1)...P(x_n \mid y = 1)}{P(X)}$$

## 3.2 Hidden Markov Model

Generative model that is mostly used for classification problems. The Hidden Markov Model is related to the Naive Bayes Classifier, in the way that, the Hidden Markov Model can be built by using the Naive Bayes Classifier on a sequence under the Markov Assumption, that he probability of a particular state is dependent only on the previous state.

## 3.3 Logistic Regression

Discriminative model that is used for classification problems.
Logistic Regression will be explained in Seminar 5 by Mathias Neitzel

## 3.4 Conditional Random Field

Discriminative model that is mostly used for classification problems. It is related to the Logistic Regression in the way that it performs analogously on sequences.

| | Single | Sequence |
|---|---|---|
| Generative | Naive Bayes Classifier | Hidden Markov Model |
| Discriminative | Logistic Regression | Conditional Radom Field |

## 3.5 Neural Networks

Neural Networks are designed to cluster raw input, recognize patterns, or interpret sensory data. Despite their multiple advantages, neural networks require significant computational resources. It is also called the 'black-box' algorithm as interpreting the logic behind their predictions can be challenging.

Neural Networks can be used for both classification as well as regression problems and can be discriminative as well as generative models.

Neural Networks will be explained in Seminar 6 by Jan Feldman.

## 3.6 Kernel Method

Class of algorithms for pattern analysis, which means finding clusters, rankings, principle components, correlations and classifications in data sets. The Kernel methods all use the "kernel trick" which computes the inner product between the images of all pairs of data, without ever computing the coordinates of the data. The "kernel trick" is used because the operation is often computationally cheaper than the explicit computation of all the coordinates.

Examples of algorithms operating with kernels:

- Support Vector Machine
- Gaussian processes
- Principal Components Analysis
- Canonical Correlation Analysis
- Ridge Regression
- Spectral Clustering

Any linear model can be turned into a non-linear model by applying the kernel trick to the model: replacing its features by a kernel function.

Kernel Methods will be explained in Seminar 7 by Leonie Pätzold.

## 3.7 Support Vector Machine

Discriminative model that is used for both classification as well as regression problems and is one the kernel methods.
A Support Vector Machine represents points in space, so that the data for the separate categories are divided by a clear gap that is as wide as possible. Predictions are based, on where the new data is mapped into space and on the side of the gap on which they fall.

## 3.8 Decision Tree

Discriminative model that is used for both classification as well as regression problems.
Tree models where the target variable can take a discrete set of values are called classification trees. In classification trees, class labels and branches represent conjunctions of features that lead to those class labels.
Decision trees where the target variable can take continuous values, typically real numbers, are called regression trees.
Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.
Decision trees are often combined with ensemble learning methods:

- Boosted Trees
  - Incrementally building an ensemble by training each new instance to emphasize the training instances previously mis-modeled
- Bootstrap Aggregated Decision Tree
  - Also called bagged decision tree, is an early ensemble method, that builds multiple decision trees by repeatedly re-sampling training data with replacement, and voting the trees for a consensus prediction
  - Random Forest classifier is a specific type of bootstrap aggregating
    * Output is the mode or mean of all the outputs of the individual trees
    * Corrects for decision trees tendency to overfit to their training set
    * Efficient for large data sets

## 3.9 Probabilistic graphical models

Probabilistic graphical models are used for both classification as well as regression problems. They all share three useful properties:

- They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
- Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph.
- Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

A probabilistic graphical model has nodes, that represent random variables and links between nodes, that express the probabilistic relationships between the nodes. Examples of probabilistic graphical models include:

- Bayesian Networks (directed graphical models)
- Markov Random Fields (undirected graphical models)

Probabilistic graphical models will be explained in Seminar 8 by Pascal Gepperth.

## 3.10 Mixture models and EM

Mixture models are used for both classification as well as regression problems. An example for mixture models are Gaussian Mixture Models, which are widely used in data mining and pattern recognition.
Traditionally Gaussian mixture models are unsupervised learning algorithms, but there are also versions that work with supervised approaches.
The expectation-maximization (EM) algorithm is a general technique for finding maximum likelihood estimators in latent variable models.

Mixture models and EM will be explained in Seminar 9 by Sascha Lill.

# 4 Additional Thoughts

## 4.1 Precision and Recall

Precision and Recall are

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

Precision (Positive Predictive Value)

$$\frac{"True\ Positives"}{"True\ Positives" + "False\ Positives"}$$

Recall (Sensitivity)

$$\frac{"True\ Positives"}{"True\ Positives" + "False\ Negatives"}$$

Examples for Precision and Recall

- Spam Detection
  - Should have high precision, because you don't want to have important emails in your spam folder. But it is OK to have some spam emails to come through
- Malicious Software
  - Should have high recall, because you have to prevent any malicious software, even if it means blocking some non malicious software

## 4.2 Selection of Features

The selection of relevant data features is essential for supervised learning to work effectively.
More features can improve accuracy, but increase computational cost.
So there is a trade off. Therefore one should prioritise features that increase accuracy the most.

## 4.3 Bias-Variance Dilemma

The bias-variance dilemma is a central problem in supervised learning. To understand it, you need to know what a bias error and a variance error is.

The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs, which is called underfitting.

The variance error is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs, which is called overfitting.

Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously.

High-variance learning methods may be able to represent their training set well but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with high bias typically produce simpler models that don't tend to overfit but may underfit their training data, failing to capture important regularities.

## 4.4 Pros and Cons of Supervised Learning

Most Pros and Cons depend on the specific techniques and algorithms implemented. Although supervised, unsupervised and reinforcement learning mostly solve different problems, there are problems that can potentially be solved by different kinds of learning. When a problem could be solved with either supervised or unsupervised the following Pros and Cons are usually correct:

Pros

- Usually more accurate and less computationally complex than unsupervised learning

Cons

- You need labeled data. This means your training set needs a corresponding target set.