Machine Learning Basics

Guido Falk von Rudorff, University of Kassel

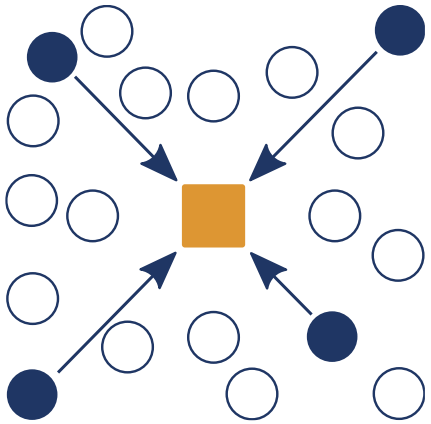
vonrudorff@uni-kassel.de nablachem.org/talks ferchault @ferchault

Goals Machine Learning

- What is ML?
- What are the key ingredients?
- Problem Classes
- Simple example Algorithms

Introduction

Machine Learning



Foundations | Statistical modelling

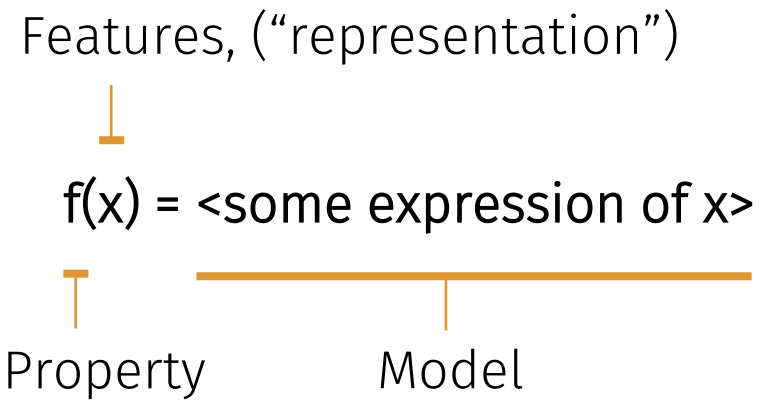
Accuracy | Systematically improvable through data and training

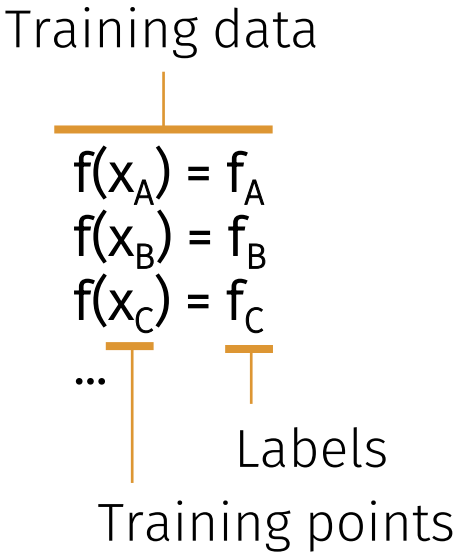
Specialty | Universal, scale-bridging, data-driven approach

Limitation | Requires training data, no black box

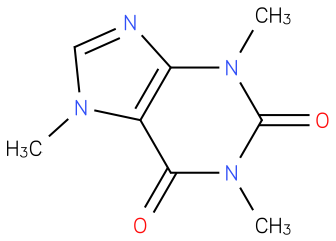
ML = Mapping compound to property using some explicit results.

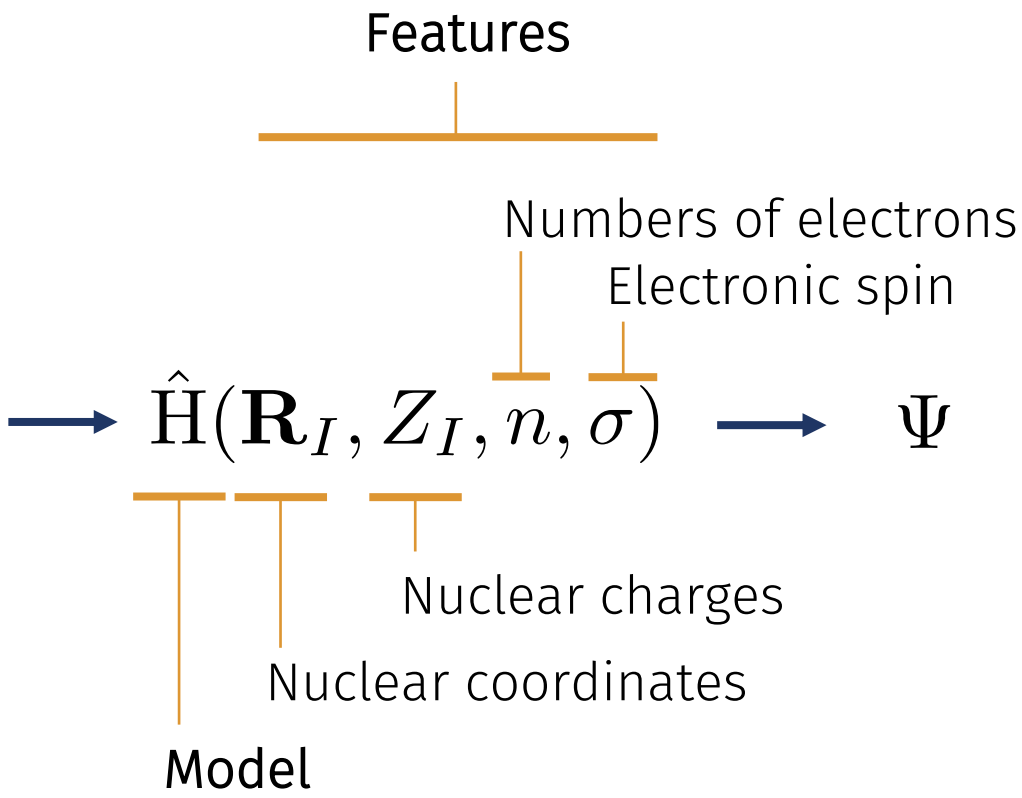
Nomenclature



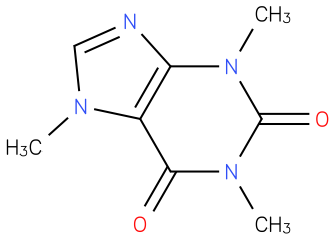


Quantum chemistry picture





Representations



0 1 0 0 1
1 1 0 1 1
1 0 1 1 1
0 1 0 0 0

Graph
Vector, Matrix, …
Bit field
String
…

Problem classes

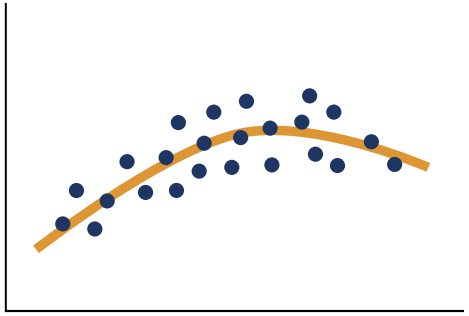
Supervised Learning
(with labels)

Unsupervised Learning
(without labels)

Classification

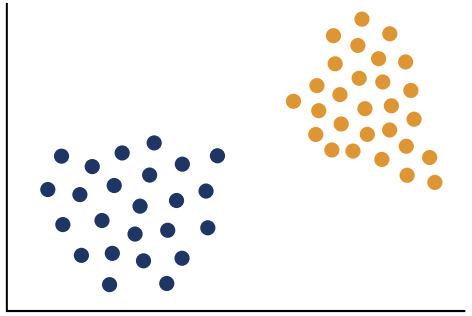
- Stability
- Reaction mechanisms
- …

Regression



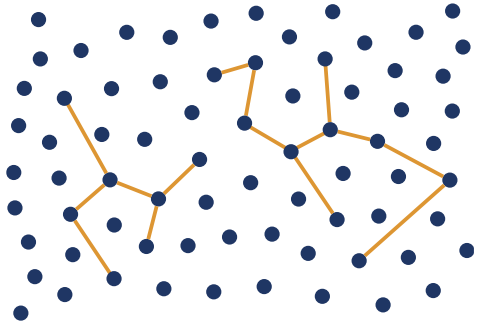
- Reaction barriers
- Geometries
- …

Clustering



- Dimensionality reduction
- Find mechanisms
- …

Association



- Find mechanisms
- Detect networks
- …

Classification

Challenges

- Imbalanced frequencies
- Irrelevant features
- Overlapping classes
- Non-linear data
- High-dimensional data

Approaches

- One vs All: n classifiers
- One vs One: n*(n-1) classifiers

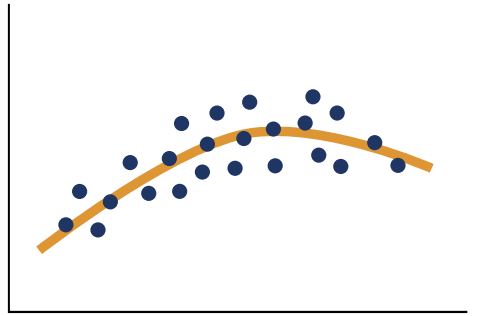
Common algorithms

- Decision trees / Random forest
- K-nearest neighbours
- Neural networks

Regression

Challenges

- Outliers
- Multicollinearity
- Non-normalised features
- Heteroscedasticity



Approaches

- Regularisation
- Bootstrapping

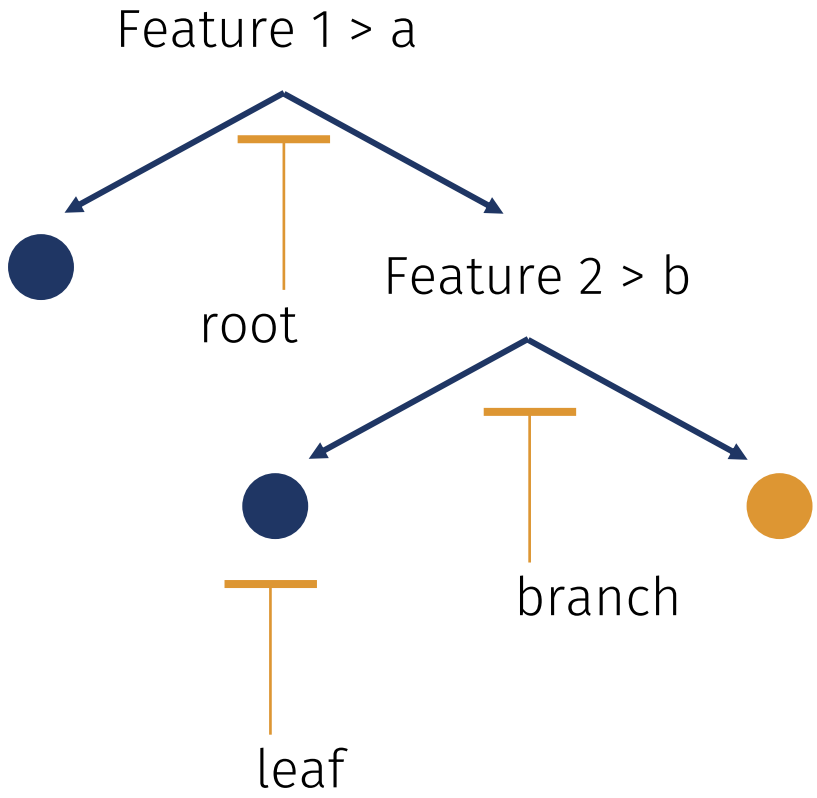
Common algorithms

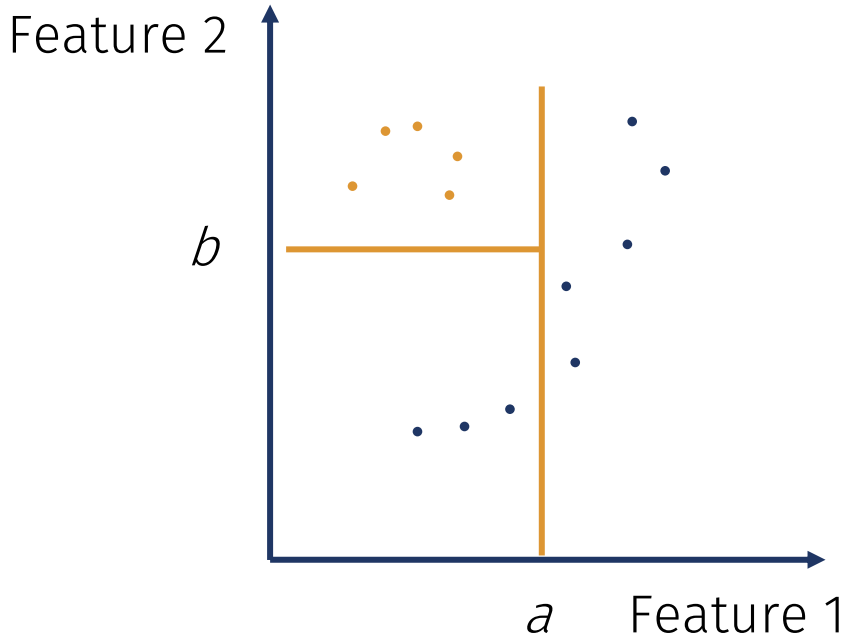
- Support vector machines
- Gaussian process regression
- Neural networks

Decision Trees

Split data along feature data range, typically binary decisions

- Scalar: find delimiter
- Discrete: yes/no





When to use

Pros

- Easy to interpret / visualize
- Cheap to use
- No data standardization necessary
- Works well with large amounts of training data (need exponentially more data for another level)
- Flexible: both regression and classification

Cons

- Costly to train
- Prone to overfitting
- Overfitting in high dimensions
- Struggles with „diagonal data“
- Struggles with imbalanced data sets
- Instable under changed of training / randomization

How to fix

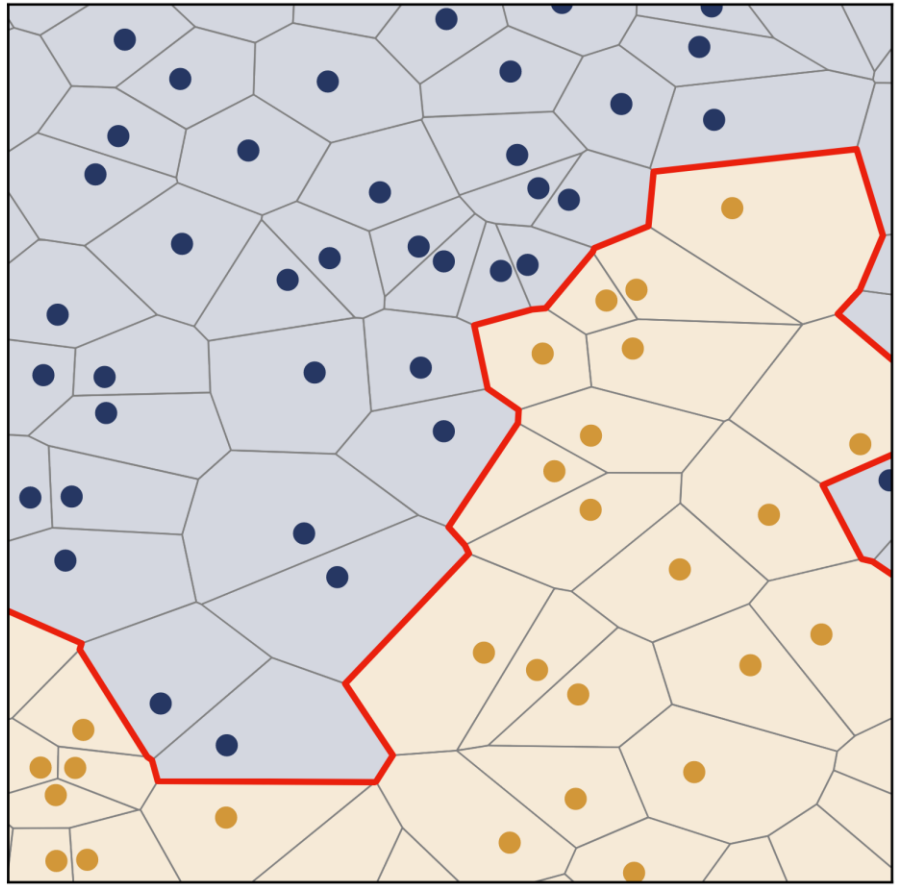
Consider random forests
Restrict depth of tree
Subselect features
Transform features with principal components
Subsample
Consider random forests

Nearest Neighbor

Idea
- Find most similar object and take its label

Requirements
- Similarity measure between objects

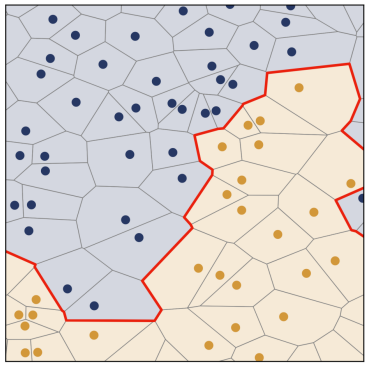
Alternatives
- Find k most similar objects and take their majority label
- Weight those similar objects by their similarity
- Transform coordinates such that classification is easier (Neighborhood component analysis)
- Transform metric (Large margin nearest neighbor)

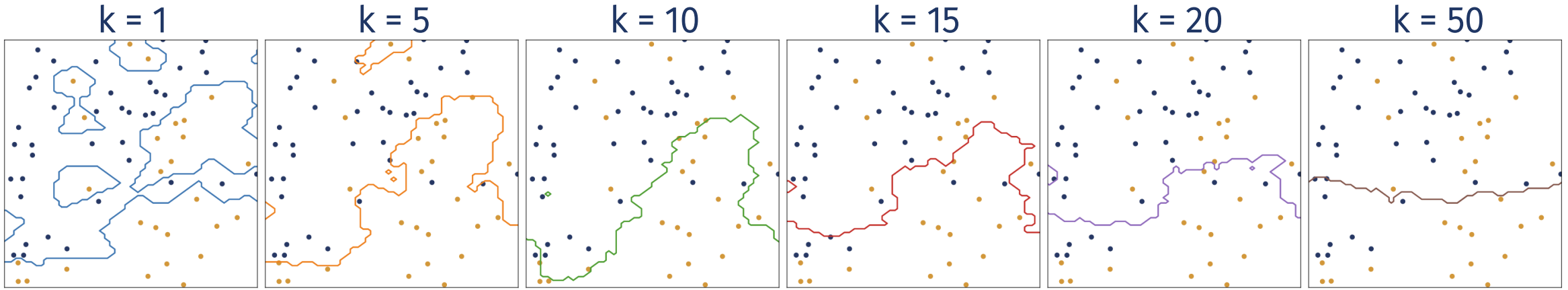


k-Nearest Neighbor

Idea

- Noisy data creates artefacts
- Find k most similar objects and take their majority label
- Small k: prone to overfit
- Large k: underfit





When to use

Pros
- No training except hyperparameter optimisation
- Simple method
- Need cheap model

Cons
- High dimensions: all points are far away from each other
- Large k needed due to noise may underfit
- Hard to query efficiently

How to fix
Project dimensions
Exploit intrinsic dimension
Change representation
Exploit static training data: index
Use approximate nearest neighbors

Summary Machine Learning

- ML: mapping input features onto output labels
- Purely data-driven, exploits similarity
- Systematic improvement of models through more data
- Not parameter-free
- Problem families: classification / regression
- Decision Tree
  - Hierarchy of conditions
  - Needs plenty of data, no closed form fit
- K-nearest neighbor
  - (Weighted) average of k most similar points
  - Needs good coverage, simple to implement
- Many other methods available