

Lecture 1: Overview

April 25, 2019

Machine Learning, Summer Term 2019

Michael Tangermann Frank Hutter Marius Lindauer

University of Freiburg



Lecture Overview

- ➊ Motivation: Why Study Machine Learning?
- ➋ Organizational Issues
- ➌ Introduction to Supervised Machine Learning
- ➍ Other Types of Machine Learning
- ➎ The Machine Learning Design Cycle
- ➏ Wrapup: Summary, Other Courses, Resources, Preview

Lecture Overview

- 1 Motivation: Why Study Machine Learning?
- 2 Organizational Issues
- 3 Introduction to Supervised Machine Learning
- 4 Other Types of Machine Learning
- 5 The Machine Learning Design Cycle
- 6 Wrapup: Summary, Other Courses, Resources, Preview

Where are you already using Machine Learning?

- You're probably using dozens of systems built on machine learning every day ...
- Let's hear some examples from you to get started! 

Annoying ads — making them less annoying
recommendation — music
search engines predicting
voice recognition — Alexa, Siri, ...
autonomous cars
image recognition
text —
medical diagnosis

Machine Learning is Everywhere

- Spam classification

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing
- Face recognition: Facebook, smart phones

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing
- Face recognition: Facebook, smart phones
- Speech recognition (Siri on the iPhone / Google talk on Android)

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing
- Face recognition: Facebook, smart phones
- Speech recognition (Siri on the iPhone / Google talk on Android)
- Machine translation (Google translate, Skype auto-translate)

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing
- Face recognition: Facebook, smart phones
- Speech recognition (Siri on the iPhone / Google talk on Android)
- Machine translation (Google translate, Skype auto-translate)
- Product/song/video recommendations in online stores

Machine Learning is Everywhere

- Spam classification
- Ranking web pages & image search: Google / Bing
- Face recognition: Facebook, smart phones
- Speech recognition (Siri on the iPhone / Google talk on Android)
- Machine translation (Google translate, Skype auto-translate)
- Product/song/video recommendations in online stores
- Computer game industry: self-configuration based on user actions

Machine Learning is Everywhere

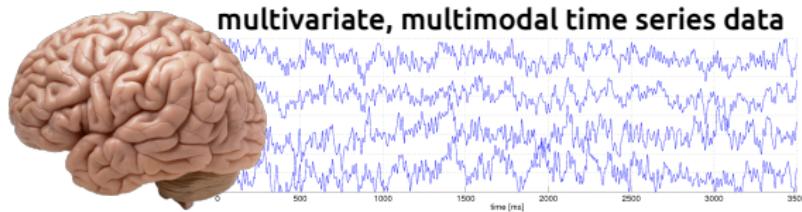
As of now, slightly less common examples:

- Autonomous cars
- Intelligent prostheses (BrainLinks-BrainTools)
- Autonomous helicopter flight
- Learning soccer robots (RoboCup)
- Learning to Design RNA

Success Story: Decoding of Brain Activity

A grand challenge! Brain signals ...

- are high-dimensional and noisy
- are subject specific
- have distributions that vary over time (fatigue, learning, strategies, ...)



Traditional way to do research:

- do the same experiment for many subjects
- average all the data
- do statistics and publish

Desired: single-trial decoding to run online applications!

Estimate Movie Content from Brain Activity



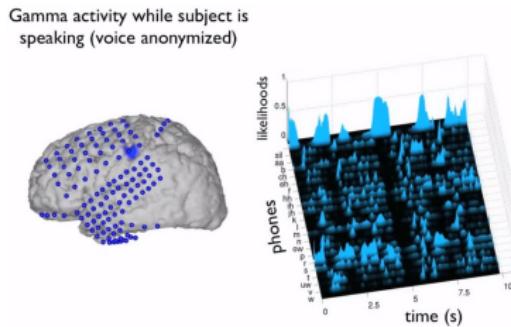
Can you guess a movie's content from brain signals of the observer?

- Show subjects several hours of movies while recording brain signals (fMRI data).
- Train a movie-to-brain activity encoding model (regression).
- Show novel movies and record brain signals.
- Estimate the visual content with the trained model.

[Nishimoto et al., 2011, Current Biology, video from

<https://www.youtube.com/watch?v=nsjDnYxJ0bo>]

Estimate Spoken Speech from Brain Activity



Reconstruct spoken speech from brain signals:

- Semi-invasive recordings
 - Train a ECoG-to-phonemes decoder
 - Use speech recognition technique + language models

[Herff et al., Front. Neurosci. 2015, <http://dx.doi.org/10.3389/fnins.2015.00217>]

Example from our work: Estimate Mental Workload

Using BCI Technology to monitor brain processes in **real time**.

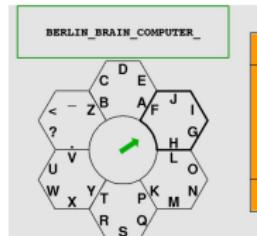
Driver's workload during:

- Driving task only (primary task)
- Driving + audio book listening
- Driving + mental calculation



Example from our work: Controlling a Brain-Computer Interface

Decode motor imagery classes (left hand, right hand, feet) from EEG signals to control a spelling application:



Today's BCI applications address:

- Communication for motor-impaired patients
- Device control
- Rehabilitation after stroke
- Mental state monitoring (workload, attention, ...)

Machine Learning and Data Mining

- Many fields have **big data** to be mined
 - Medicine, biology, astronomy, finance, social network studies, . . .

Machine Learning and Data Mining

- Many fields have **big data** to be mined
 - Medicine, biology, astronomy, finance, social network studies, ...
- We'd like to make inferences from the data
 - “We're drowning in data but starving for knowledge”
(Futurist John Naisbitt)
 - Machine Learning lets us **generalize** (which is key to knowledge)

Machine Learning and Artificial Intelligence

- Machine learning is a subfield of AI

Machine Learning and Artificial Intelligence

- Machine learning is a subfield of AI
- AI used to be focused on **logics**
 - Theorem proving
 - Classical planning

Machine Learning and Artificial Intelligence

- Machine learning is a subfield of AI
- AI used to be focused on **logics**
 - Theorem proving
 - Classical planning
- Since the 1990s, AI focuses more on **probabilistic reasoning**
 - This is closer to machine learning

Machine Learning and Artificial Intelligence

- Machine learning is a subfield of AI
- AI used to be focused on **logics**
 - Theorem proving
 - Classical planning
- Since the 1990s, AI focuses more on **probabilistic reasoning**
 - This is closer to machine learning
- One approach to solve AI is to build machine learning systems that **resemble the human brain**
 - Richard Feynman: “what we can’t synthesize, we don’t understand”
 - Deep neural networks are a (small) step in this direction

Machine Learning as A Different Way of Programming

- We don't understand how the human brain solves certain problems
 - Face recognition
 - Playing Atari games
 - Speech recognition
 - Picking the next move in the game of Go
- We can nevertheless learn these tasks from data/experience

Machine Learning as A Different Way of Programming

- We don't understand how the human brain solves certain problems
 - Face recognition
 - Playing Atari games
 - Speech recognition
 - Picking the next move in the game of Go
- We can nevertheless learn these tasks from data/experience
- If the task changes, we simply re-train

Machine Learning as A Different Way of Programming

- We don't understand how the human brain solves certain problems
 - Face recognition
 - Playing Atari games
 - Speech recognition
 - Picking the next move in the game of Go
- We can nevertheless learn these tasks from data/experience
- If the task changes, we simply re-train
- We can construct computer systems that are too complex for us to understand anymore ourselves...
 - E.g., deep neural networks have millions of weights.
 - E.g., AlphaGo, the system that beat world champion Lee Sedol
 - + David Silver, lead author of AlphaGo cannot explain its moves
 - + Paraphrased: "You would have to ask a Go expert."

Machine Learning is very Sought After

- AI is very popular
 - When university students in a wide range of fields were asked to list a field they would like to study other than their own, the most frequent answer was AI
 - Machine learning is one of the most exciting subfields of AI

Machine Learning is very Sought After

- AI is very popular
 - When university students in a wide range of fields were asked to list a field they would like to study other than their own, the most frequent answer was AI
 - Machine learning is one of the most exciting subfields of AI
- Predictions by a 2011 McKinsey report (for the US):
 - Better data analysis could save the public sector 250 billion Euro/year
 - There will be a need for 140,000 to 190,000 machine learning experts
 - There will be a need for 1.5 million data-savvy managers

Lecture Overview

- 1 Motivation: Why Study Machine Learning?
- 2 Organizational Issues
- 3 Introduction to Supervised Machine Learning
- 4 Other Types of Machine Learning
- 5 The Machine Learning Design Cycle
- 6 Wrapup: Summary, Other Courses, Resources, Preview

Organization: Instructors

Michael Tangermann

Research group leader
Brain State Decoding
Main Point of Contact



Marius Lindauer

Junior Research Group Leader
Machine Learning



Frank Hutter

Professor
Machine Learning



Organization: Assistants and Tutors



Sebastián
Castaño



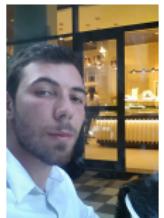
David
Hübner



Jan
Sosulski



Andreas
Meinel



Arber
Zela



Katharina
Eggensperger



Simon
Ging



Yi-Chun
Lin



Baohe
Zhang

Organization: Lectures

- Meeting times:
Monday & Thursday 16:15-17:52 (S101, room 00 026)
 - Including a 7-minute break roughly in the middle
- Both Monday and Thursday slots might have lecture content and exercise discussion
 - We highly recommend that you attend both lectures & exercise sessions
- All lectures will be recorded & online soon after
 - Thanks to our Hiwis Simon, Yi-Chun & Bahoe

Organization: Lectures

- Meeting times:
Monday & Thursday 16:15-17:52 (S101, room 00 026)
 - Including a 7-minute break roughly in the middle
- Both Monday and Thursday slots might have lecture content and exercise discussion
 - We highly recommend that you attend both lectures & exercise sessions
- All lectures will be recorded & online soon after
 - Thanks to our Hiwis Simon, Yi-Chun & Bahoe
- We strive for interaction!
 - Discuss with your neighbor:  (2 minutes)
 - Raise your hands to give answers: 
 - Multiple-choice questions  to be answered with colours on smartphone screens
 - + E.g., "Color Screen" on iPhone, or go to:
 - + <https://ml.informatik.uni-freiburg.de/teaching/colors.html>

Organization: Assignments

- They are voluntary: we don't want to police you
- You can work in teams of up to 3 members

Organization: Assignments

- They are voluntary: we don't want to police you
- You can work in teams of up to 3 members
- With 200 students we cannot provide much individual feedback
 - Hiwis will provide some remarks on submissions

Organization: Assignments

- They are voluntary: we don't want to police you
- You can work in teams of up to 3 members
- With 200 students we cannot provide much individual feedback
 - Hiwis will provide some remarks on submissions
- We'll try to optimize amount learned per time spent with assignments
 - Proposals for improvements are always welcome
- Keeping up with the assignments is the best preparation for the exam

Organization: Gitlab and ILIAS

- Gitlab Server
 - A private Git repository just for you (your team)
 - Whatever is committed by the deadline is your submission

- Gitlab Server

- A private Git repository just for you (your team)
- Whatever is committed by the deadline is your submission

- ILIAS

- All lecture materials will be posted in ILIAS
(incl. assignments, code to get you started on assignments, etc)
- There is a forum for questions of all kinds
 - + Making a first post there is part of Assignment 1
 - + Please answer a question if you know the answer
- There is also a Wiki

Organization: Amount of Work / ECTS points

- This course yields 6 ECTS points → 180 working hours
- Lectures (\approx 45 hours) + assignments + exam preparation
- Time management options
 - A 6-8 hours per assignment, little exam prep. RECOMMENDED
 - B 4-5 hours per assignment, much exam prep. MINIMUM
 - C 0 hours per assignment, ??? exam prep. VERY BAD IDEA

Organization: Amount of Work / ECTS points

- This course yields 6 ECTS points → 180 working hours
- Lectures (\approx 45 hours) + assignments + exam preparation
- Time management options
 - A 6-8 hours per assignment, little exam prep. RECOMMENDED
 - B 4-5 hours per assignment, much exam prep. MINIMUM
 - C 0 hours per assignment, ??? exam prep. VERY BAD IDEA
- There is a written exam in the end
 - We'll make a mock exam available about two weeks before
 - We're trying to fix the exam date soon (likely mid August)
 - + Erasmus students: same exam as every one else
 - Doing all the exercise sheets and understanding everything behind them is the perfect preparation for the exam

Lecture Overview

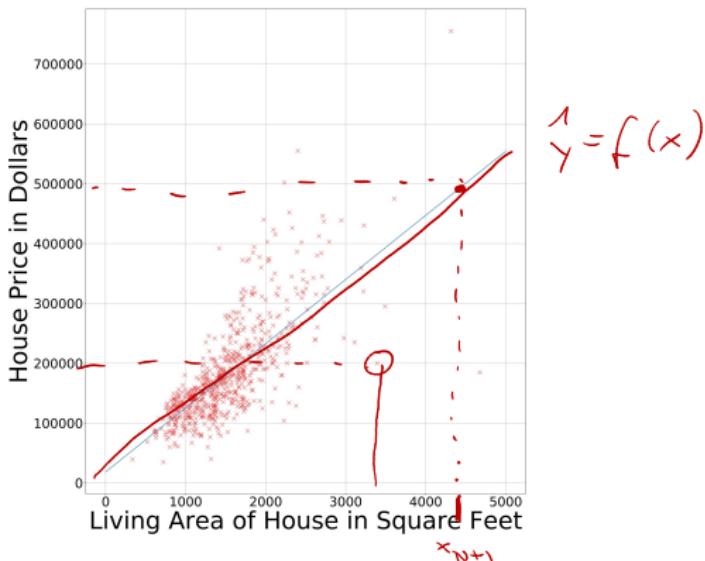
- 1 Motivation: Why Study Machine Learning?
- 2 Organizational Issues
- 3 Introduction to Supervised Machine Learning
- 4 Other Types of Machine Learning
- 5 The Machine Learning Design Cycle
- 6 Wrapup: Summary, Other Courses, Resources, Preview

Supervised Learning: The Basic Idea

- Use past experience to predict the future
 - Use **labelled data points** $\langle(\underline{\mathbf{x}}_i, \underline{y}_i)\rangle_{i=1}^N$ that we collected in the past
 - to automatically construct a model whose prediction $\hat{\underline{y}}_{N+1}$ for a new data point $\underline{\mathbf{x}}_{N+1}$ is close to the actual label \underline{y}_{N+1} .

Supervised Learning: The Basic Idea

- Use past experience to predict the future
 - Use **labelled data points** $\langle(\underline{x}_i, \underline{y}_i)\rangle_{i=1}^N$ that we collected in the past
 - to automatically construct a model whose prediction \hat{y}_{N+1} for a new data point \underline{x}_{N+1} is close to the actual label y_{N+1} .
- First example: predicting house price based on size (in square feet)



Supervised Learning: The Basic Idea

- Use past experience to predict the future
 - Use **labelled data points** $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$ that we collected in the past
 - to automatically construct a model whose prediction \hat{y}_{N+1} for a new data point \mathbf{x}_{N+1} is close to the actual label y_{N+1} .
- Machine learning terminology:
 - Data point \mathbf{x}_i , often a vector in \mathbb{R}^D
 - Label y_i
 - + **Regression:** $y_i \in \mathbb{R}$
 - + **Classification:** y_i discrete, e.g. $y_i \in \{\text{true, false}\}$, or $y_i \in \{\text{German, English, Spanish}\}$

Supervised Learning: The Basic Idea

- Use past experience to predict the future
 - Use **labelled data points** $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$ that we collected in the past
 - to automatically construct a model whose prediction \hat{y}_{N+1} for a new data point \mathbf{x}_{N+1} is close to the actual label y_{N+1} .
- Machine learning terminology:
 - Data point \mathbf{x}_i , often a vector in \mathbb{R}^D
 - Label y_i
 - + **Regression:** $y_i \in \mathbb{R}$
 - + **Classification:** y_i discrete, e.g. $y_i \in \{\text{true, false}\}$, or $y_i \in \{\text{German, English, Spanish}\}$
 - Past experience = training set $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$
 - Automatically construct = learn, fit, induce
 - Model = function, hypothesis, classifier/regressor

Supervised Learning: The Basic Idea

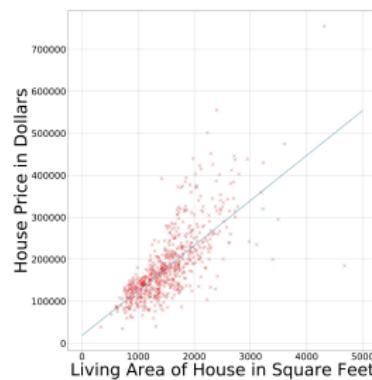
- Use past experience to predict the future
 - Use labelled data points $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$ that we collected in the past
 - to automatically construct a model whose prediction \hat{y}_{N+1} for a new data point \mathbf{x}_{N+1} is close to the actual label y_{N+1} .
- Machine learning terminology:
 - Data point $\underline{\mathbf{x}}_i$, often a vector in \mathbb{R}^D
 - Label y_i
 - + Regression: $y_i \in \mathbb{R}$
 - + Classification: y_i discrete, e.g. $y_i \in \{\text{true, false}\}$, or $y_i \in \{\text{German, English, Spanish}\}$
 - Past experience = training set $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^N$
 - Automatically construct = learn, fit, induce
 - Model = function, hypothesis, classifier/regressor
- What is the dimensionality D in the house price example?
 - ★ D=1 ✓
 - ★ D=2
 - ★ D=3
 - ★ D=4

Supervised Learning: A First Regression Example

Predicting housing prices

- For many houses in a training set, we know:
 - the square footage
 - the prize they sold for
- We'd like to predict the prize for houses based on their square footage
- One data point: square footage x_i and its prize y_i (in US\$)
- This is a regression problem since $y_i \in \mathbb{R}$

Square footage x_i	Prize y_i
1.710	208.500
1.262	181.500
1.786	223.500
1.717	140.000
2.198	250.000
1.362	143.000
1.694	307.000
2.090	200.000
1.774	129.900
1.077	118.000
...	...



Supervised Learning: a Bit More Standard Notation

- We learn a function f that maps each **data point** \mathbf{x}_i to a prediction \hat{y}_i that resembles the true y_i

Supervised Learning: a Bit More Standard Notation

- We learn a function f that maps each **data point** \mathbf{x}_i to a prediction \hat{y}_i that resembles the true y_i
- In the previous example, $\underline{\mathbf{x}}_i$ only had one dimension, but it will typically be a (column) vector with D dimensions
 - That's why it is bold-faced
 - The dimensions of $\underline{\mathbf{x}}_i$ are $\underline{D \times 1}$
 - The dimensions of its transpose \mathbf{x}_i^T are $1 \times D$

$$\begin{pmatrix} \vdots \\ i \end{pmatrix}$$

Supervised Learning: a Bit More Standard Notation

- We learn a function f that maps each **data point** \mathbf{x}_i to a prediction \hat{y}_i that resembles the true y_i
- In the previous example, \mathbf{x}_i only had one dimension, but it will typically be a (column) vector with D dimensions
 - That's why it is bold-faced
 - The dimensions of \mathbf{x}_i are $D \times 1$
 - The dimensions of its transpose \mathbf{x}_i^T are $1 \times D$
- We stack all N data points on top of each other:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \dots \\ \mathbf{x}_N^T \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix}$$

$N \times D$ $N \times 1$

Supervised Learning: a Bit More Standard Notation

- We learn a function f that maps each **data point** \mathbf{x}_i to a prediction \hat{y}_i that resembles the true y_i
- In the previous example, \mathbf{x}_i only had one dimension, but it will typically be a (column) vector with D dimensions
 - That's why it is bold-faced
 - The dimensions of \mathbf{x}_i are $D \times 1$
 - The dimensions of its transpose \mathbf{x}_i^T are $1 \times D$
- We stack all N data points on top of each other:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

- Thus, **\mathbf{X}** is a $N \times D$ matrix; **\mathbf{y}** is a $N \times 1$ matrix
 - \mathbf{y} is bold-faced since it's a vector
 - \mathbf{X} is capital and bold-faced since it's a matrix
- **\mathbf{y}** are the 'correct' labels; this is our **supervisory** signal (thus the name)

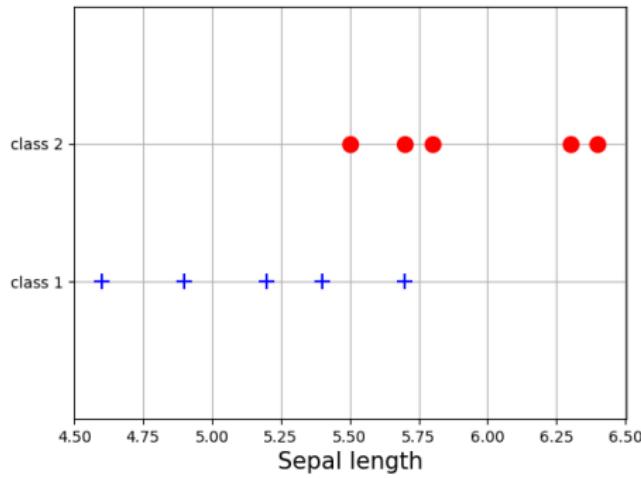
Supervised Learning: a Simple Classification Example

- A classical data set from Botany: classifying Iris flowers
 - feature 1: sepal length
 - feature 2: sepal width



$x_{i,1}$	$x_{i,2}$	y_i
6.40	2.90	2
5.50	2.50	2
5.20	3.50	1
4.60	3.60	1
5.70	3.80	1
6.30	2.50	2
5.80	2.60	2
4.90	3.10	1
5.70	2.80	2
5.40	3.90	1

$X \quad 10 \times 2 \quad Y$



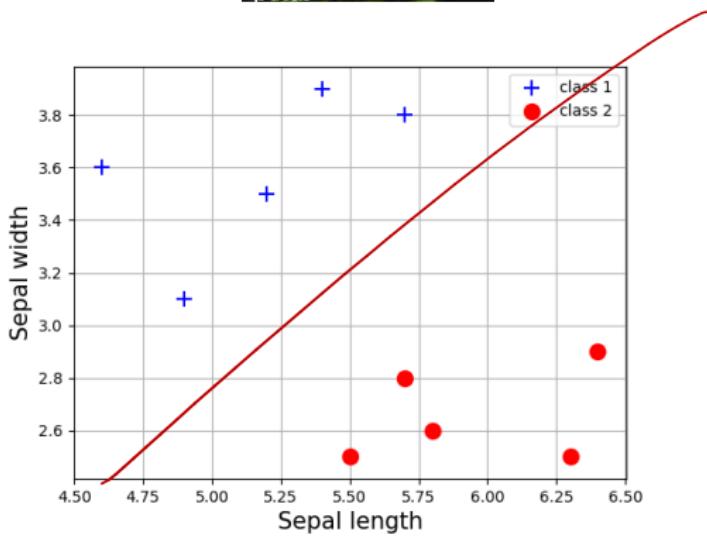
Supervised Learning: a First Classification Example

- A classical data set from Botany: classifying Iris flowers
 - feature 1: sepal length
 - feature 2: sepal width



$x_{i,1}$	$x_{i,2}$	y_i
6.40	2.90	2
5.50	2.50	2
5.20	3.50	1
4.60	3.60	1
5.70	3.80	1
6.30	2.50	2
5.80	2.60	2
4.90	3.10	1
5.70	2.80	2
5.40	3.90	1

$N \times D$



Mini-Quiz in Preparation for Assignment 1: Iris Data

- Classifying Iris flowers (the Iris data set)
 - You will learn a classifier based on 100 labelled Iris flowers
 - Iris flowers are described by 4 characteristics (sepal length/width, petal length/width)
 - There are 3 different classes of Iris flowers

Mini-Quiz in Preparation for Assignment 1: Iris Data

- Classifying Iris flowers (the Iris data set)
 - You will learn a classifier based on 100 labelled Iris flowers
 - Iris flowers are described by 4 characteristics (sepal length/width, petal length/width)
 - There are 3 different classes of Iris flowers
- The standard supervised learning setup
 - You learn a classifier based on \mathbf{X}_{train} and \mathbf{y}_{train}
 - You use that classifier to predict \mathbf{y}_{test} given \mathbf{X}_{test}

$$\begin{matrix} \left[\begin{matrix} \mathbf{y}_{N+1} \\ \vdots \\ \mathbf{y}_{N+n} \end{matrix} \right] & \left[\begin{matrix} \mathbf{x}_{N+1} \\ \vdots \\ \mathbf{x}_{N+m} \end{matrix} \right] \\ \mathbf{y}_{test} & \mathbf{x}_{test} \end{matrix}$$

Mini-Quiz in Preparation for Assignment 1: Iris Data

- Classifying Iris flowers (the Iris data set)
 - You will learn a classifier based on 100 labelled Iris flowers
 - Iris flowers are described by 4 characteristics (sepal length/width, petal length/width)
 - There are 3 different classes of Iris flowers
- The standard supervised learning setup
 - You learn a classifier based on \mathbf{X}_{train} and \mathbf{y}_{train}
 - You use that classifier to predict \mathbf{y}_{test} given \mathbf{X}_{test}
- What is the correct dimensionality for the matrix \mathbf{X}_{train} ?
 - ★ 100×3
 - ★ 3×4
 - ★ 100×4 ✓
 - ★ 4×100

Supervised Learning Example: Credit Risk

- A private customer comes to a bank and applies for a loan
- Should the bank give him the loan?
 - We can build a decision support system for the bank employee

Supervised Learning Example: Credit Risk

- A private customer comes to a bank and applies for a loan
- Should the bank give him the loan?
 - We can build a decision support system for the bank employee
 - What features could the employee use?



income

age

→ credit history

savings

assets

purpose of loan

crime history

current employment

marital status

Supervised Learning Example: Credit Risk

- A private customer comes to a bank and applies for a loan
- Should the bank give him the loan?
 - We can build a decision support system for the bank employee
 - What features could the employee use? 
 - Features: income, age, credit amount, etc.

Supervised Learning Example: Credit Risk

- A private customer comes to a bank and applies for a loan
- Should the bank give him the loan?
 - We can build a decision support system for the bank employee
 - What features could the employee use? 
 - Features: income, age, credit amount, etc.
- Ways of casting this as a machine learning problem:
 - 1. Should the bank give the loan?
 - 2. How much interest should the bank charge to break even in expectation?

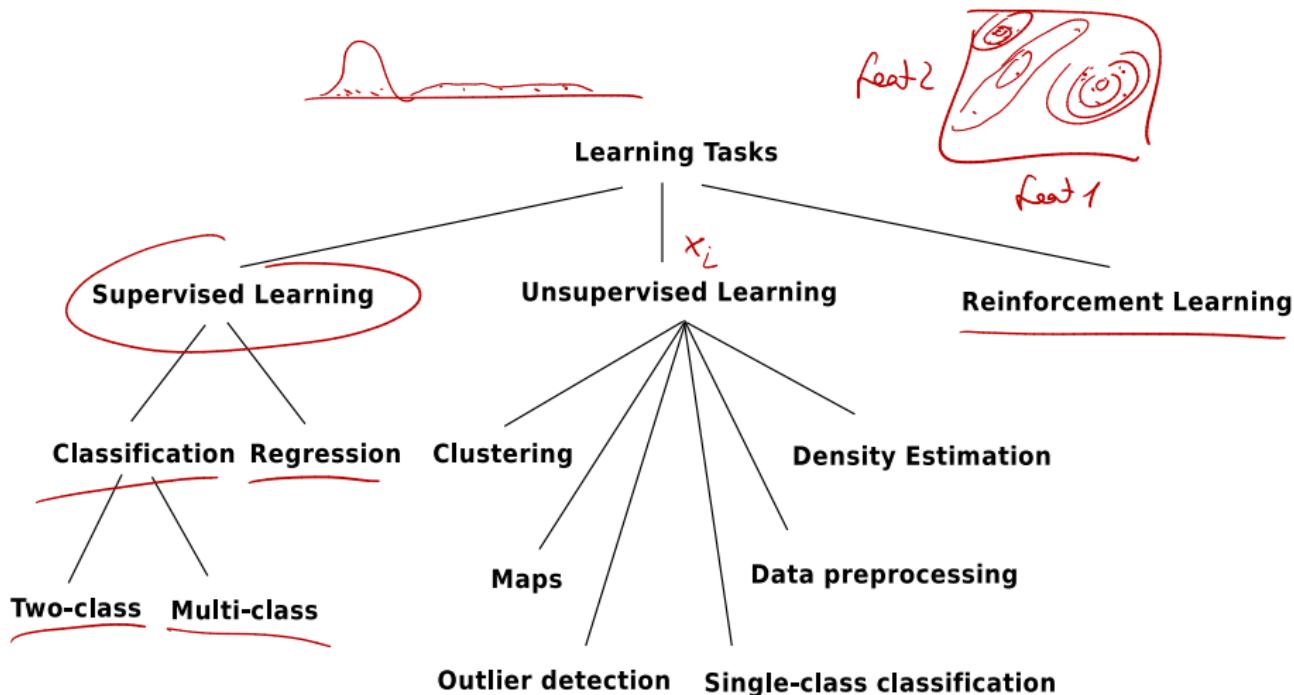
Supervised Learning Example: Credit Risk

- A private customer comes to a bank and applies for a loan
- Should the bank give him the loan?
 - We can build a decision support system for the bank employee
 - What features could the employee use? 
 - Features: income, age, credit amount, etc.
- Ways of casting this as a machine learning problem:
 - 1. Should the bank give the loan?
 - 2. How much interest should the bank charge to break even in expectation?
- Please raise the color of the correct statement:
 - ★ Both are regression tasks.
 - ★ Both are classification tasks.
 - ★ 1. is regression, 2. is classification.
 - ★ 1. is classification, 2. is regression. ✓

Lecture Overview

- 1 Motivation: Why Study Machine Learning?
 - 2 Organizational Issues
 - 3 Introduction to Supervised Machine Learning
 - 4 Other Types of Machine Learning
 - 5 The Machine Learning Design Cycle
 - 6 Wrapup: Summary, Other Courses, Resources, Preview
- .

Types of Learning Tasks



Types of learning tasks: Unsupervised Learning

- training data consists of a description of situations/ objects/ ...
- learn what is typical/ similar/ interesting/ strange within the data
- no teacher
- example: **clustering** of things that 'somehow belong together'
 - e.g. cluster customers into those with similar behavior
- example: **novelty detection**
 - detect objects, that are surprising / different from others
 - e.g. find suspicious outliers in radioactivity measurements

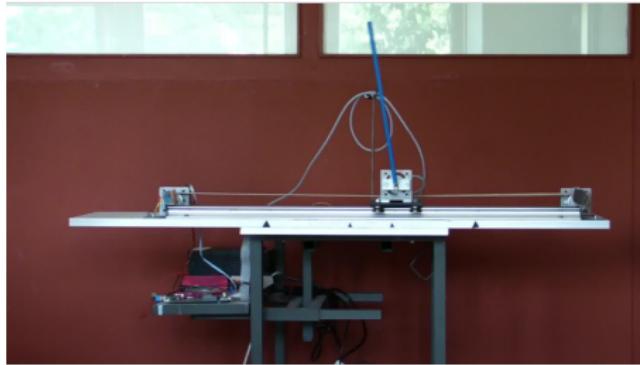
Types of learning tasks: Reinforcement learning

- training data consist of sequences of situations, actions and reward
- task is: learn a control strategy to maximize the reward
- no teacher
- (famous) example: learn to balance a pole

→ treated in detail in lecture [Reinforcement Learning](#)

Example from our work: learning dynamics models for optimal control

- task is to swing up and balance a pole on a cart
- no knowledge of system dynamics
- use of regression technique to learn dynamics function from (random) interaction
- use model for planning locally optimal trajectories (model-based RL)



Example from our work: algorithm runtime prediction

- Task: predict how long an algorithm will run
 - on new inputs
 - with new parameter settings
- regression problem

Example from our work: algorithm runtime prediction

- Task: predict how long an algorithm will run
 - on new inputs
 - with new parameter settings
- regression problem
- Task: will the algorithm terminate within 1 minute?
- classification problem

Example from our work: algorithm runtime prediction

- Task: predict how long an algorithm will run
 - on new inputs
 - with new parameter settings
- regression problem
- Task: will the algorithm terminate within 1 minute?
- classification problem
- Are there different types of input instances?
- clustering problem

Example from our work: algorithm runtime prediction

- Task: predict how long an algorithm will run
 - on new inputs
 - with new parameter settings
- regression problem
- Task: will the algorithm terminate within 1 minute?
- classification problem
- Are there different types of input instances?
- clustering problem
- Can we adapt the algorithm's parameters at runtime to make it better?
- reinforcement learning

Lecture Overview

- 1 Motivation: Why Study Machine Learning?
- 2 Organizational Issues
- 3 Introduction to Supervised Machine Learning
- 4 Other Types of Machine Learning
- 5 The Machine Learning Design Cycle
- 6 Wrapup: Summary, Other Courses, Resources, Preview

Applying machine learning in practice

- Let's say we want to apply machine learning to a practical problem
- What do you think are some of the things we need to do? 

Gather data

train/test split

decide on model

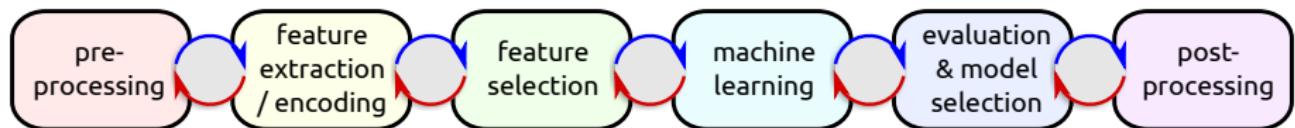
removing incorrect/incomplete data

decide on type of ML task

scope definition - what does the user want

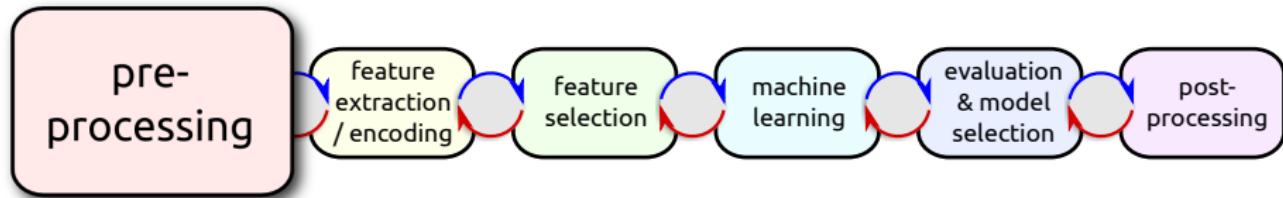
dimensions & characteristics of the data

The Machine Learning Design Cycle



- Real ML applications have many different steps, including:
 - Getting the data into the convenient (X,y) form
 - Casting your problem as an ML problem (supervised, unsupervised, RL)
 - Evaluating your solution & using it in practice

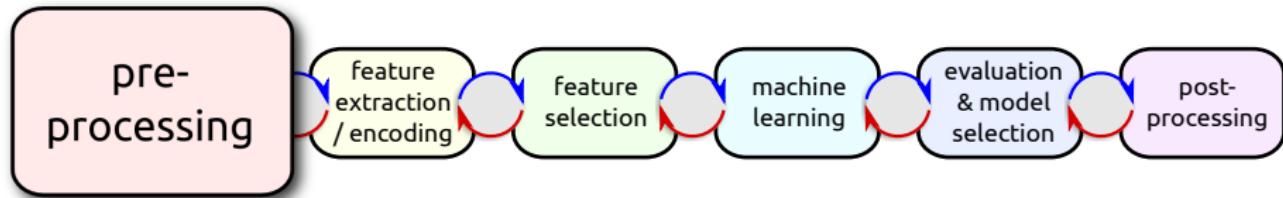
ML Design Cycle: Preprocessing



Get the data in the first place

- Design sensor systems, design questionnaires
- Pay people for labels
- Get legal approval to use data, etc.

ML Design Cycle: Preprocessing



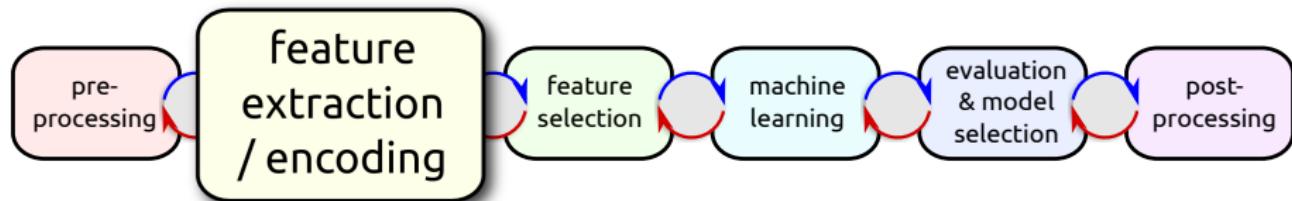
Get the data in the first place

- Design sensor systems, design questionnaires
- Pay people for labels
- Get legal approval to use data, etc.

Clean the data

- Missing values may not be missing at random
- Outliers may be due to errors in data acquisition or carry meaning
- Imbalanced data → e.g., subsample majority class / create additional data from minority class, etc.

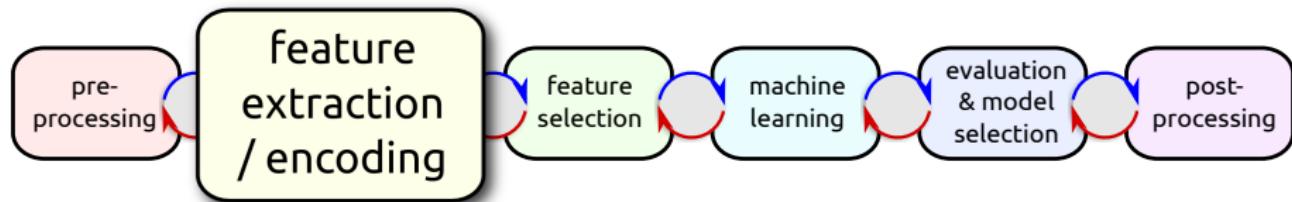
ML Design Cycle: Feature Extraction & Encoding (1/3)



Extract features: getting a nice X matrix

- Often unclear what the features should be.
 - E.g., what's a good feature representation of a human cell, a text, a sound, a brainwave, an image?
- Solution 1: use domain knowledge!
- Solution 2: representation learning (e.g., deep neural networks)

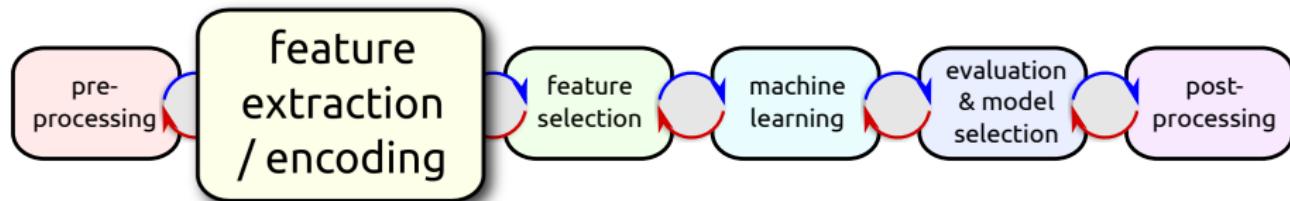
ML Design Cycle: Feature Extraction & Encoding (2/3)



Combine features:

- We can use complex operations to combine features
- E.g., body mass index: $BMI = \text{weight} / (\text{height}^2)$
(With weight in kg and height in cm)
- Combined features can be more expressive than their components

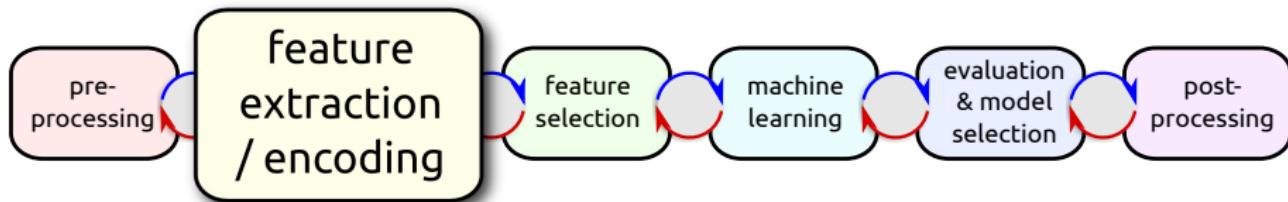
ML Design Cycle: Feature Extraction & Encoding (3/3)



Encode features: getting to a standard format

- Most prominent example: **categorical features**
 - Features with unordered, finite domain
 - E.g., {red, green, blue}
 - These could be coded as red=1, green=2, blue=3
 - But red is not closer to green than it is to blue ...

ML Design Cycle: Feature Extraction & Encoding (3/3)

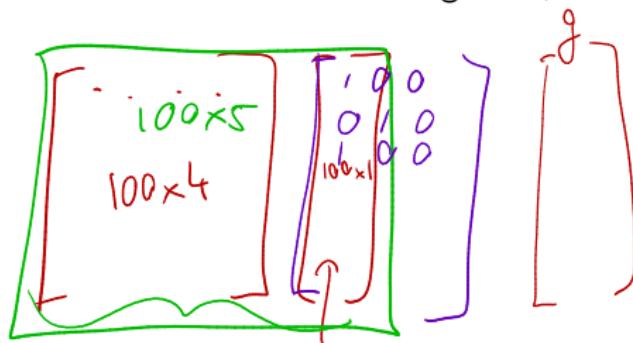


Encode features: getting to a standard format

- Most prominent example: **categorical features**
 - Features with unordered, finite domain
 - E.g., {red, green, blue}
 - These could be coded as red=1, green=2, blue=3
 - But red is not closer to green than it is to blue ...
- Standard solution: **one-hot-encoding** (aka 1-in-k encoding)
 - k -ary feature is coded as k features in $\{0,1\}$ with exactly one 1
 - E.g., red = [1,0,0]
 - E.g., green = [0,1,0]
 - E.g., blue = [0,0,1]

Mini-Quiz in Preparation for Assignment 1: Iris Data

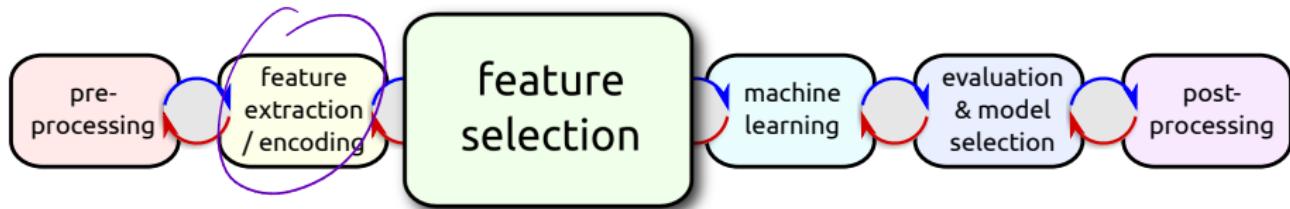
- Classifying Iris Flowers
 - You will learn a classifier based on 100 labelled Iris flowers
 - Iris flowers are described by 4 characteristics (sepal length/width, petal length/width)
 - There are 3 different classes of Iris flowers
- Let's say we make an additional feature out of the class of flowers (and want to predict whether goats like the flower or not)
 - The flower class is categorical, so we'll use a one-hot encoding



Mini-Quiz in Preparation for Assignment 1: Iris Data

- Classifying Iris Flowers
 - You will learn a classifier based on 100 labelled Iris flowers
 - Iris flowers are described by 4 characteristics (sepal length/width, petal length/width)
 - There are 3 different classes of Iris flowers
- Let's say we make an additional feature out of the class of flowers (and want to predict whether goats like the flower or not)
 - The flower class is categorical, so we'll use a one-hot encoding
- What is the correct dimensionality for the matrix \mathbf{X}_{train} now?
 - ★ Still 100×4
 - ★ 100×5
 - ★ 100×6
 - ★ 100×7 ✓

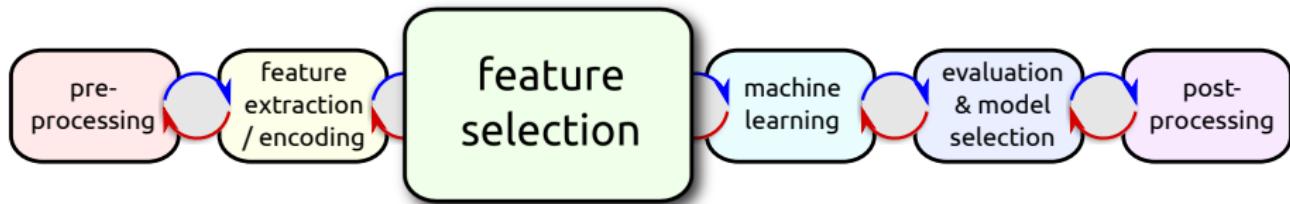
ML Design Cycle: Feature Selection



Select features: getting rid of distractions

- Sometimes, we have too many features (D features)
 - Source 1: original data (e.g., 20,000-25,000 human protein-coding genes)
 - Source 2: feature creation (e.g., trying lots of feature combinations)

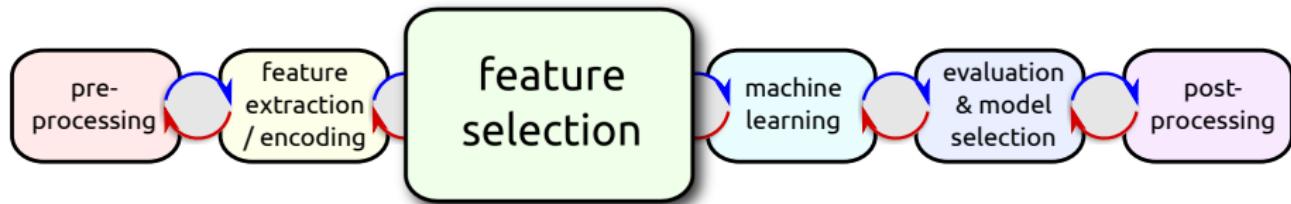
ML Design Cycle: Feature Selection



Select features: getting rid of distractions

- Sometimes, we have too many features (D features)
 - Source 1: original data (e.g., 20,000-25,000 human protein-coding genes)
 - Source 2: feature creation (e.g., trying lots of feature combinations)
- We need to select the ones useful for the machine learner
 - Strongly depends on the learning algorithm applied
 - Some algorithms do 'automatic' feature selection, scale to large D

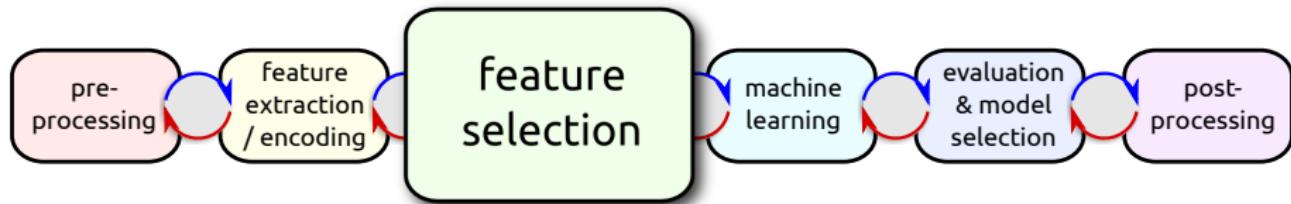
ML Design Cycle: Feature Selection



Select features: getting rid of distractions

- Sometimes, we have too many features (D features)
 - Source 1: original data (e.g., 20,000-25,000 human protein-coding genes)
 - Source 2: feature creation (e.g., trying lots of feature combinations)
- We need to select the ones useful for the machine learner
 - Strongly depends on the learning algorithm applied
 - Some algorithms do 'automatic' feature selection, scale to large D
 - Some algorithms' computational complexity scales poorly in D
 - + They simply run out of memory when D is too large

ML Design Cycle: Feature Selection

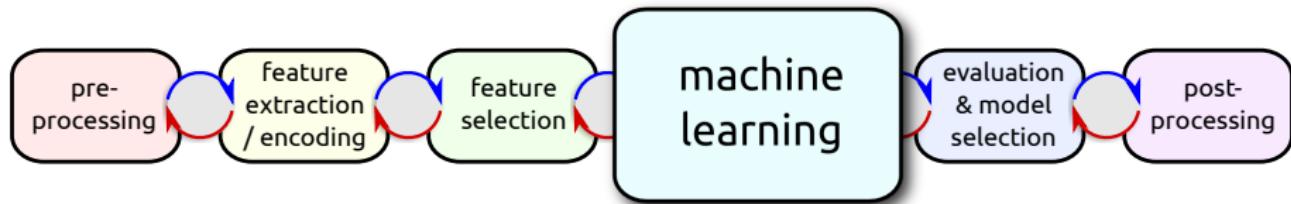


Select features: getting rid of distractions

$N \neq \text{data}$
 $D \# \text{features}$

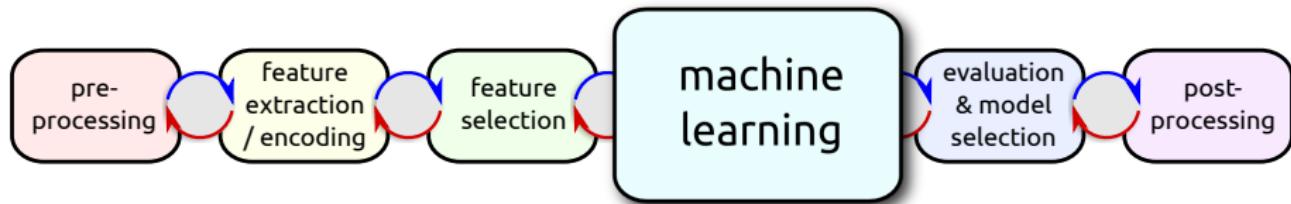
- Sometimes, we have too many features (D features)
 - Source 1: original data (e.g., 20,000-25,000 human protein-coding genes)
 - Source 2: feature creation (e.g., trying lots of feature combinations)
- We need to select the ones useful for the machine learner
 - Strongly depends on the learning algorithm applied
 - Some algorithms do 'automatic' feature selection, scale to large D
 - Some algorithms' computational complexity scales poorly in D
 - + They simply run out of memory when D is too large
 - Some algorithms' predictive performance scales poorly in D
 - + They won't break but simply won't do very well

ML Design Cycle: 'Machine Learning'



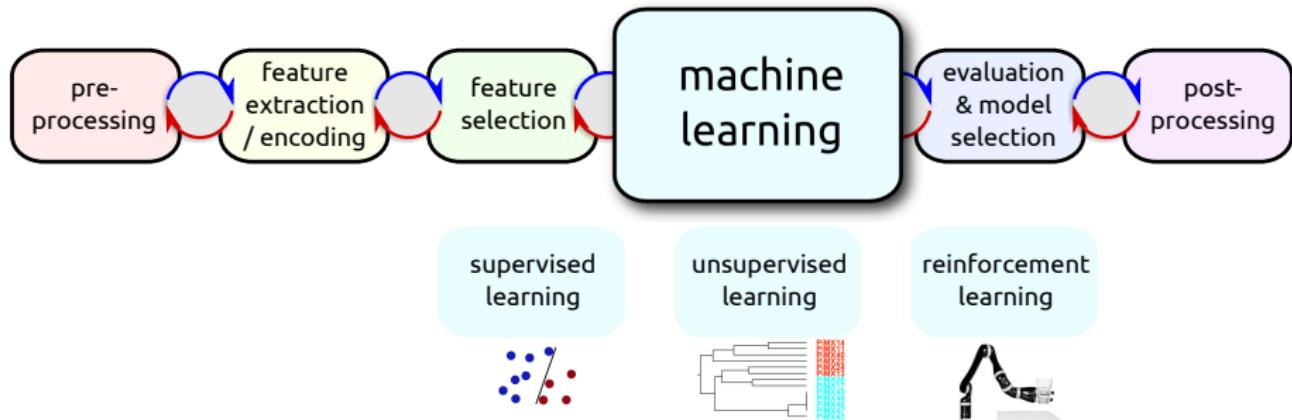
- The 'core' of a machine learning system
- Different algorithms we could use to solve a problem
- This is where 90% of machine learning research happens
 - Computational efficiency
 - Statistical efficiency
 - Provable properties

ML Design Cycle: 'Machine Learning'



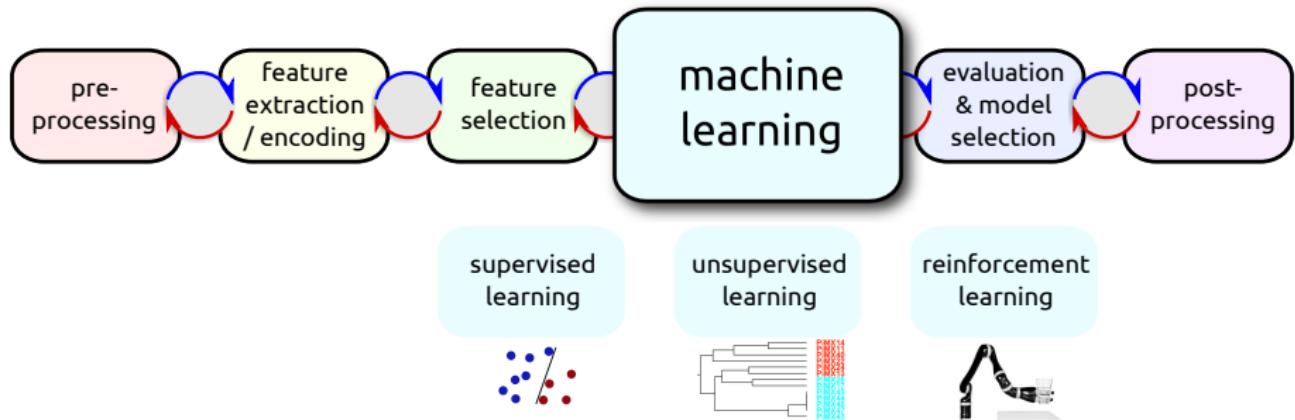
- The 'core' of a machine learning system
- Different algorithms we could use to solve a problem
- This is where 90% of machine learning research happens
 - Computational efficiency
 - Statistical efficiency
 - Provable properties
- In practice, this is not much more important than the other parts
- We'll still focus on this part (since there is the most 'meat')
 - But we connect to the other parts wherever possible
 - In practice, the features are often more important than the algorithm

ML Design Cycle: 'Machine Learning'



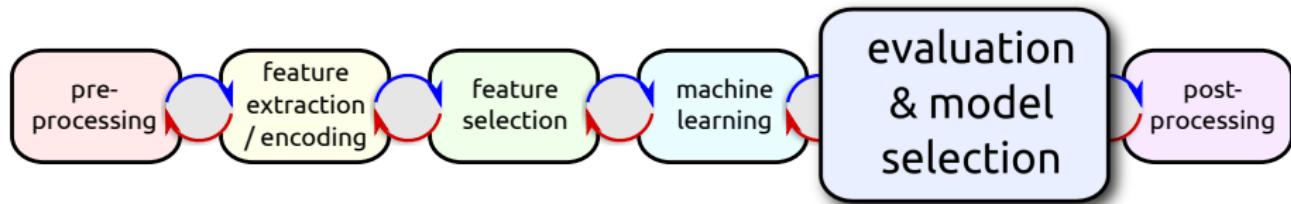
- Sometimes, you can phrase your learning problem in various ways
- Or you can combine various machine learning paradigms
- E.g., AlphaGo:
 - classification to predict expert moves
 - regression to evaluate board positions
 - reinforcement learning by self-play

ML Design Cycle: 'Machine Learning'



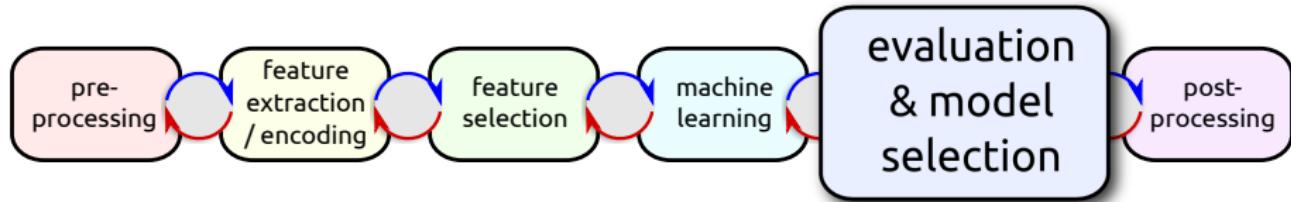
- E.g., learning in infants
 - Lots of unsupervised data
 - + Continuous perception of the environment
 - + Can suffice to learn a representation
 - Very few data points with a supervision signal y
 - + E.g., 'Look, that's called a dog!'
 - Infants can learn the class 'dog' from a handful of examples

ML Design Cycle: Evaluation and Model Selection 1/3



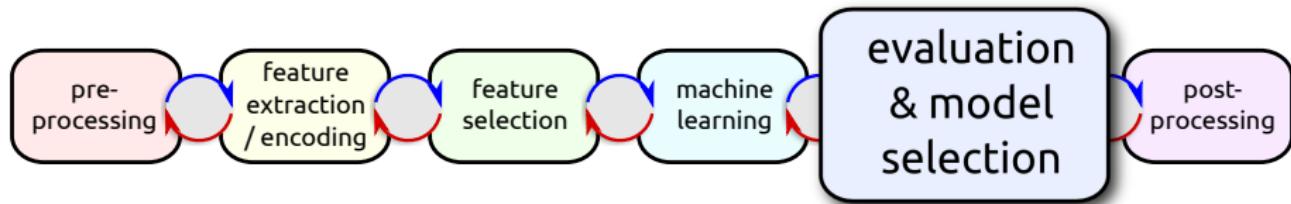
Evaluation:

- We want our models to **generalize**
 - I.e., perform well on previously unseen data points
- What does it mean to perform well?
 - Precision, recall, false positives, false negatives
 - Speed at training time, speed at test time, memory, accuracy



Model Selection: Which Algorithm to Use?

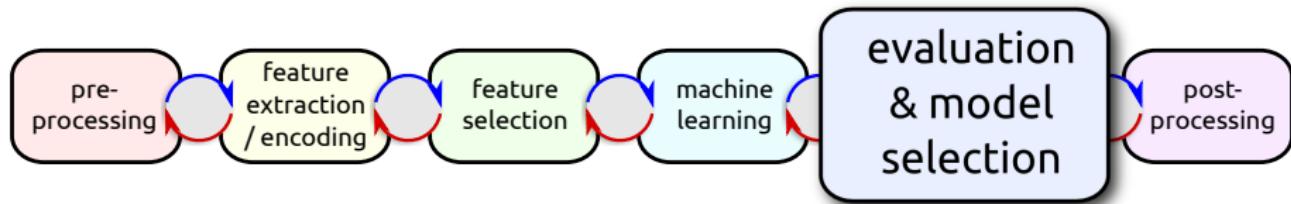
- Computational efficiency, accuracy
- Estimates of generalization performance
- Simplest approach: split dataset into training and validation set
 - Choose algorithm that does best on the validation set
 - Only that algorithm gets deployed and applied to new test data



Model Selection: Which Algorithm to Use?

- Computational efficiency, accuracy
- Estimates of generalization performance
- Simplest approach: split dataset into training and validation set
 - Choose algorithm that does best on the validation set
 - Only that algorithm gets deployed and applied to new test data
 - To evaluate on a fixed dataset
 - + Split dataset into training/validation/test
 - + Lock away test set for final test
 - + Train on training part, choose best w.r.t. validation part

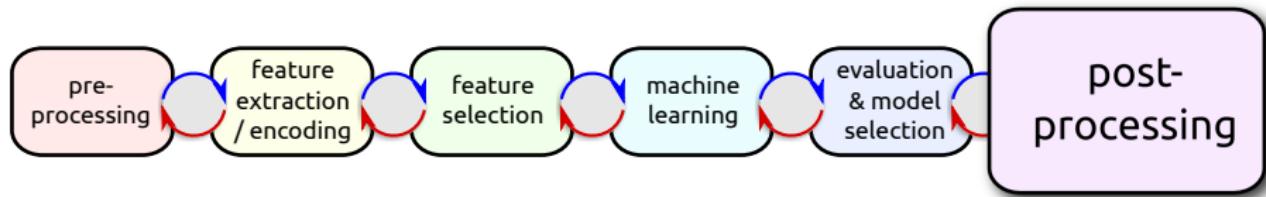
ML Design Cycle: Evaluation and Model Selection 3/3



Model Selection: How to set free parameters of the algorithm?

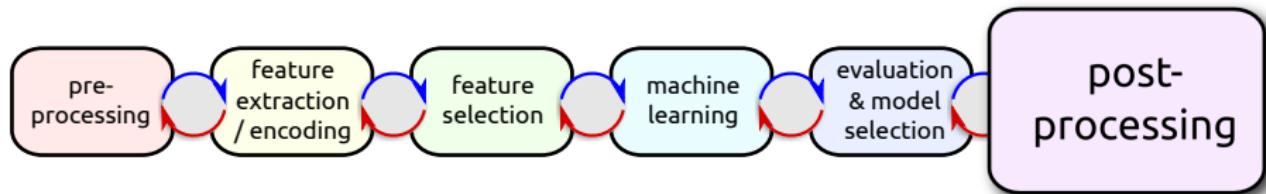
- In machine learning, free tuning parameters are called **hyperparameters**
 - in contrast to internal **model parameters**
- Find the hyperparameter setting with best validation performance
- Exponential number of possible settings

The Machine Learning Design Cycle



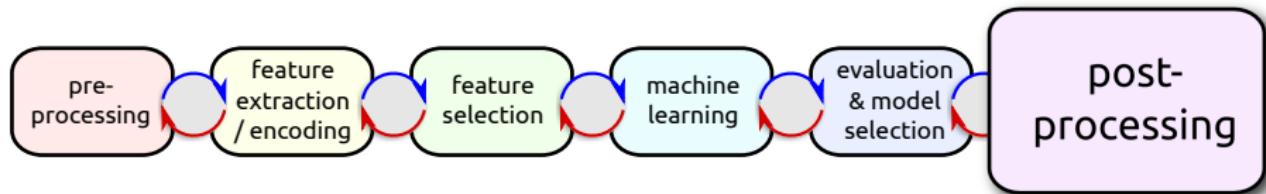
- Ensure **fairness** / avoid algorithmic bias
 - E.g., model should not discriminate based on race, gender, religion, sexual orientation, ...

The Machine Learning Design Cycle



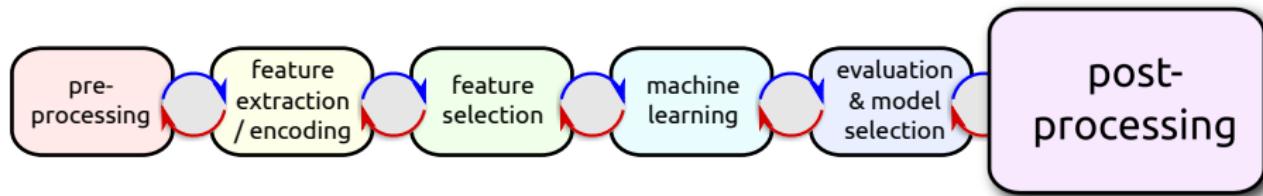
- Ensure **fairness** / avoid algorithmic bias
 - E.g., model should not discriminate based on race, gender, religion, sexual orientation, ...
- Some models don't yield proper probabilities
 - Transform model outputs to probabilities

The Machine Learning Design Cycle



- Ensure **fairness** / avoid algorithmic bias
 - E.g., model should not discriminate based on race, gender, religion, sexual orientation, ...
- Some models don't yield proper probabilities
 - Transform model outputs to probabilities
- Integrate model predictions with other sources of uncertainty
 - E.g., prediction whether a treatment would help a patient
 - E.g., prediction of side effects
 - E.g., prediction whether the ward has a free bed ...

The Machine Learning Design Cycle



- Ensure **fairness** / avoid algorithmic bias
 - E.g., model should not discriminate based on race, gender, religion, sexual orientation, ...
- Some models don't yield proper probabilities
 - Transform model outputs to probabilities
- Integrate model predictions with other sources of uncertainty
 - E.g., prediction whether a treatment would help a patient
 - E.g., prediction of side effects
 - E.g., prediction whether the ward has a free bed ...
- Deploy the model we found to achieve something
 - Don't just predict, but **act** based on the model

Lecture Overview

- 1 Motivation: Why Study Machine Learning?
- 2 Organizational Issues
- 3 Introduction to Supervised Machine Learning
- 4 Other Types of Machine Learning
- 5 The Machine Learning Design Cycle
- 6 Wrapup: Summary, Other Courses, Resources, Preview

Summary by learning goals

Having heard this lecture, you can now ...

- explain why so many people are interested in machine learning
- give examples for successful machine learning applications
- describe supervised machine learning
- distinguish between classification and regression problems
- distinguish supervised machine learning from other machine learning problems
- describe the parts of the machine learning pipeline

Most related courses at Uni Freiburg

- Statistical pattern recognition

- By Thomas Brox, [this term](#)
- A probabilistic machine learning course
- Our course is designed to be complementary
- If your main interest is machine learning,
we strongly encourage you to take this course as well

Most related courses at Uni Freiburg

- Statistical pattern recognition

- By Thomas Brox, [this term](#)
- A probabilistic machine learning course
- Our course is designed to be complementary
- If your main interest is machine learning,
we strongly encourage you to take this course as well

- Foundations of Deep Learning

- By Frank Hutter Joschka Boedecker, WS
- More advanced course that builds on this current course

Most related courses at Uni Freiburg

- Statistical pattern recognition

- By Thomas Brox, [this term](#)
- A probabilistic machine learning course
- Our course is designed to be complementary
- If your main interest is machine learning,
we strongly encourage you to take this course as well

- Foundations of Deep Learning

- By Frank Hutter Joschka Boedecker, WS
- More advanced course that builds on this current course

- Reinforcement Learning

- By Joschka Boedecker, WS
- More advanced course that builds on this current course

Most related courses at Uni Freiburg

- Statistical pattern recognition
 - By Thomas Brox, [this term](#)
 - A probabilistic machine learning course
 - Our course is designed to be complementary
 - If your main interest is machine learning,
we strongly encourage you to take this course as well
- Foundations of Deep Learning
 - By Frank Hutter Joschka Boedecker, WS
 - More advanced course that builds on this current course
- Reinforcement Learning
 - By Joschka Boedecker, WS
 - More advanced course that builds on this current course
- Lab Course Deep Learning
 - By groups Boedecker, Brox, Burgard, Hutter, SS
 - More advanced course that builds on the Deep Learning course

Other related courses at Uni Freiburg

- Artificial Intelligence

- By groups Burgard, Nebel, Boedecker, Hutter
- This term

Other related courses at Uni Freiburg

- Artificial Intelligence

- By groups Burgard, Nebel, Boedecker, Hutter
- This term

- Information Retrieval

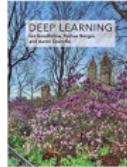
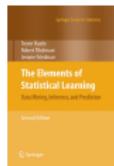
- By Hannah Bast, WS

Other related courses at Uni Freiburg

- Artificial Intelligence
 - By groups Burgard, Nebel, Boedecker, Hutter
 - This term
- Information Retrieval
 - By Hannah Bast, WS
- Automated Machine Learning (AutoML)
 - By Marius Lindauer & Frank Hutter, SS
 - More advanced course that builds on this current course & Deep Learning

We'll use material from several books, including these

- The Elements of Statistical Learning
by Hastie, Tibshirani and Friedman
 - Available online for free: <https://web.stanford.edu/~hastie/ElemStatLearn/>
- Pattern Classification by Duda, Hart and Stork
- Deep Learning
by Goodfellow, Bengio and Courville
 - HTML version available online for free:
<http://www.deeplearningbook.org/>



Python tutorials

- General:
 - Official tutorial:
<https://docs.python.org/3/tutorial/index.html>
 - For beginners: www.learnpython.org/
 - For programmers:
http://stephensugden.com/crash_into_python/
 - Many more: <http://docs.python-guide.org/en/latest/intro/learning/>
- Libraries:
 - Numpy: http://wiki.scipy.org/Tentative_NumPy_Tutorial
 - SciPy: <http://docs.scipy.org/doc/scipy/reference/tutorial/>
 - Matplotlib: <http://matplotlib.org/users/beginner.html>
- Feel free to discuss about these and additional ones on ILIAS ...

Other resources for the course

We'll also refer to other sources for background throughout. Here is one to start with:

- Khan Academy: <https://www.khanacademy.org/>
 - Useful to review linear algebra, etc
- Again, please post resources you find useful on ILIAS

Preview of Assignment 1

- Set up infrastructure for the remainder of the course
- Explore a simple classification dataset
- Classify with some manually-derived rules

Preview of Assignment 1

- Set up infrastructure for the remainder of the course
- Explore a simple classification dataset
- Classify with some manually-derived rules
- Preparation **to do today**: download our virtual machine (VM) from
<http://ml.informatik.uni-freiburg.de/~zelaa/ml2019/mlss19.ova>
 - The VM has 2GB, so you cannot download it on the fly on Monday
 - Instructions for how to set it up are on
<http://ml.informatik.uni-freiburg.de/~zelaa/ml2019>, but we'll also cover this on Monday

Preview of Assignment 1

- Set up infrastructure for the remainder of the course
- Explore a simple classification dataset
- Classify with some manually-derived rules
- Preparation [to do today](#): download our virtual machine (VM) from <http://ml.informatik.uni-freiburg.de/~zelaa/ml2019/mlss19.ova>
 - The VM has 2GB, so you cannot download it on the fly on Monday
 - Instructions for how to set it up are on <http://ml.informatik.uni-freiburg.de/~zelaa/ml2019>, but we'll also cover this on Monday
- On Monday:
 - Bring your (charged!) laptops with the VM image on it!
 - We'll set up the teams
 - Space-permitting, we'll arrange seating so we can walk through & help

